INTO THE LIDARVERSE: A REAL-TIME UNIFIED LIDAR WORLD MODEL ACROSS TASKS AND SENSORS

Anonymous authors
Paper under double-blind review

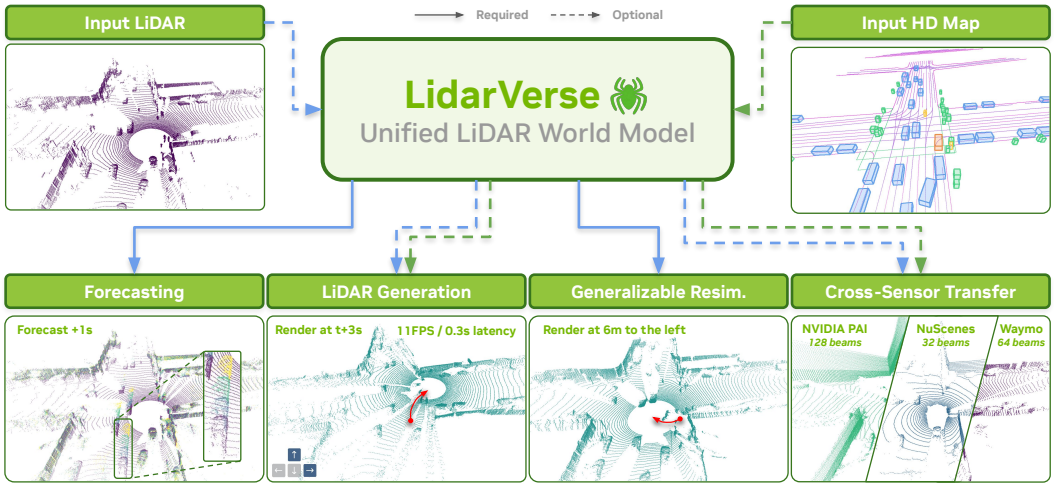

Figure 1: **LidarVerse** unifies generative and reconstruction-based LiDAR sensor simulation. By changing conditions, one distilled causal checkpoint generates new worlds, forecasts future point clouds, or resimulates observed scenes along new viewpoints with real-time, long-horizon inference.

ABSTRACT

We present LidarVerse, a real-time unified LiDAR world model for sensor simulation across tasks and sensors. Existing LiDAR simulators largely follow one of two complementary paradigms: reconstruction-based methods preserve the geometry of a captured scene but require lengthy per-scene optimization, whereas generative methods amortize learning across data but lack the consistency needed for long-horizon rollouts. LidarVerse brings both paradigms into a single, unified conditional generative framework. A universal tokenizer maps LiDAR range sequences into a shared latent space, where a bidirectional teacher learns a common interface for conditioning on a LiDAR prefix, an HD-Map, a persistent spatial cache, and optionally, text. Varying the available scene evidence turns the same model into a generator of new worlds, a point-cloud forecaster, or a generalizable resimulator of observed scenes without per-scene optimization. We distill the teacher into a four-step autoregressive causal student and maintain both a rolling temporal K/V cache and a world-aligned spatial cache, enabling persistent long-horizon rollouts with real-time inference at 11 FPS on a single H100 GPU. LidarVerse achieves state-of-the-art performance on most reported metrics across all three evaluation protocols, while its tokenizer supports 32-, 64-, and 128-beam LiDAR and generalizes to an unseen scanning pattern. These results demonstrate that generative and reconstruction-grounded LiDAR simulation can be realized within one model, providing the core capabilities for future closed-loop sensor simulation. Anonymous project page: https://anonymous11693.github.io/website/.

1 INTRODUCTION

Sensor simulation is critical for developing autonomous systems. Robots and self-driving vehicles must operate safely across a long tail of environments, trajectories, and interactions that cannot be collected exhaustively in the real world. Recent reconstruction-based systems transform captured driving logs into interactive simulation environments, while generative simulators synthesize

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.

controllable observations beyond the captured data. Together, they highlight two complementary capabilities: preserving an observed world while changing the sensor trajectory, and creating new worlds beyond recorded experience. LiDAR is especially important in this setting because it provides direct geometric measurements under challenging illumination, yet its sparse, viewpoint-dependent returns are difficult to simulate faithfully. A practical LiDAR simulator should offer both capabilities while producing accurate observations, remaining temporally coherent under sensor motion, and running fast enough for interactive use.

Neural LiDAR simulation has largely developed along the same divide. *Reconstruction-based* methods fit an explicit neural scene representation to a captured log and render LiDAR from novel sensor poses (Wu et al., 2024; Huang et al., 2023; Zheng et al., 2024; Zhou et al., 2025a). However, fitting each scene can require hours of optimization, and the resulting simulator remains tied to a manually collected log. These methods can vary the sensor trajectory within a scene, but do not readily scale to a diverse distribution of new environments. *Generative* methods instead amortize scene modeling across data and learn a distribution from which novel LiDAR scenes can be sampled (Zyrianov et al., 2025; Zhou et al., 2025b; Liang et al., 2026). However, LiDAR generation still lacks a compact latent interface that scales across both time and sensors. As a result, existing generators tend to specialize in one sensor configuration, produce only short sequences, or require many denoising steps. The field thus lacks a unified simulator that combines the distributional coverage of generative models with the persistence and control of reconstruction-based methods.

We address this gap with LidarVerse, a real-time unified LiDAR world model that brings generative and reconstruction-based simulation into one conditional framework. Its foundation is a universal, fully attention-based LiDAR tokenizer designed specifically for range images, capable of reconstructing data from 32-, 64-, and 128-beam sensors and generalizing to an unseen scanning pattern. On this latent space, we train a bidirectional diffusion teacher with a unified interface for a LiDAR prefix, an HD-Map and actor layout, a persistent spatial cache (PSC), and optionally text. Rather than assigning each task a separate architecture or prediction head, we sample these conditions independently during training and use their composition to specify the inference regime. Generation, forecasting, and resimulation are therefore different ways of querying the same learned world, distinguished by the available evidence rather than by model parameters.

To make this unified model practical for interactive simulation, we distill the bidirectional teacher into a four-step block-autoregressive causal student. A rolling temporal K/V cache retains recent dynamics, while a world-aligned PSC preserves scene geometry beyond the finite temporal window, enabling persistent and temporally coherent long-horizon rollouts. The resulting causal checkpoint retains the shared conditioning interface and performs all three tasks at or above the native 10-Hz LiDAR rate. Across their respective evaluation protocols, LidarVerse outperforms task-specific baselines on most reported metrics. Together, these results establish a unified sensor-simulation paradigm in which changing the conditions, rather than the architecture, moves one model between generative simulation and reconstruction-grounded resimulation. Our main contributions are:

- **Universal LiDAR tokenization.** We introduce a fully attention-based tokenizer that preserves native temporal resolution and improves geometric reconstruction and generative modeling.
- **Unified LiDAR sensor simulation.** We bridge generative and reconstruction-based simulation within one conditional world model supporting multiple tasks.
- **Real-time, persistent world modeling.** We distill the bidirectional teacher into a real-time causal student and introduce a rolling PSC for consistent long-horizon generation.

2 Related Work

LiDAR Generation Early LiDAR generation methods focus on synthesizing individual sweeps in point, voxel, BEV, or range-image representations (Zyrianov et al., 2022; Xiong et al., 2023; Nakashima & Kurazume, 2024; Nakashima et al., 2025; Ran et al., 2024; Hu et al., 2024). LiDARGen (Zyrianov et al., 2022) pioneered diffusion-based range-image synthesis, establishing a strong foundation for data-driven LiDAR simulation. Subsequent methods extended synthesis beyond individual sweeps to temporally coherent sequences or 3D/4D geometry from which LiDAR can be rendered (Zyrianov et al., 2025; Li et al., 2025; Ren et al., 2024; Xu et al., 2026; Ren et al., 2025). More recently, LiDARCrafter (Liang et al., 2026) and LaGen (Zhou et al., 2025b) formulate LiDAR sequence generation autoregressively, while Sensor2Sensor (Wang et al., 2026) autoregressively

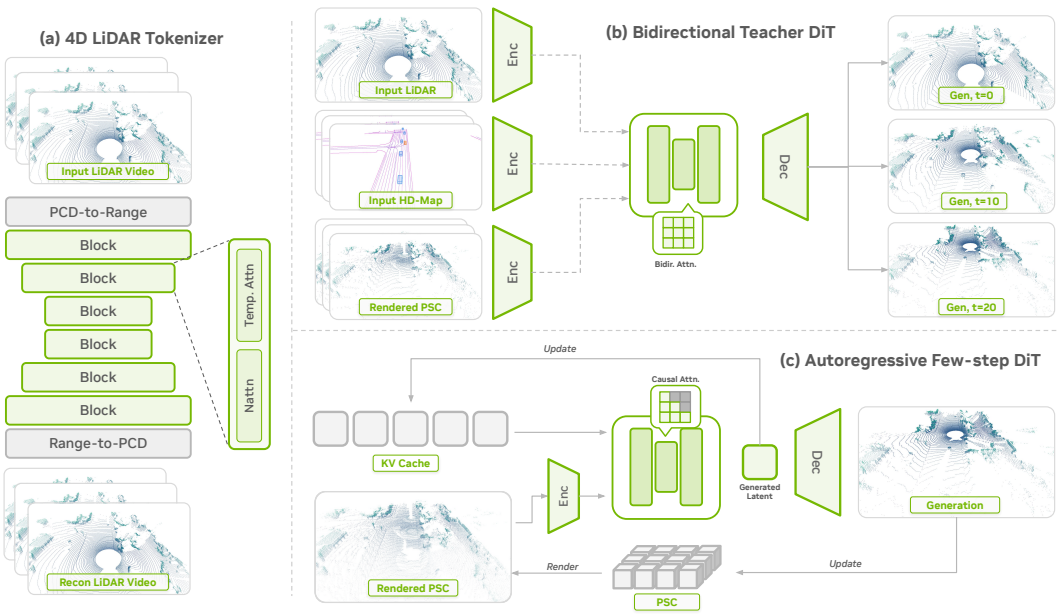

Figure 2: **Architecture**: LidarVerse is trained in three stages: (a) An attention-based LiDAR tokenizer learns a latent space while preserving spatial and temporal structure (Sec. 3.1). (b) A bidirectional teacher learns a unified interface for conditioning on a LiDAR prefix, an HD-Map, a persistent spatial cache, and optionally, text (Sec. 3.2). (c) Causal distillation and a rolling persistent spatial cache enable real-time, long-horizon inference with the same conditioning interface (Sec. 3.3).

converts monocular video into multimodal driving logs. However, these methods are limited to short rollouts and do not maintain consistency over long horizons. In contrast, LidarVerse combines few-step self-forcing distillation with persistent 3D memory, maintaining coherent long rollouts while generating at the native sensor rate in real time.

LiDAR Resimulation LiDAR resimulation reconstructs a captured scene and renders it along novel sensor trajectories (Yang et al., 2023; Tonderski et al., 2024; Wu et al., 2024; Tao et al., 2023; Zheng et al., 2024; Zhou et al., 2025a; Turki et al., 2026). Most existing approaches rely on per-scene optimization, which requires lengthy reconstruction and tends to overfit the recorded trajectory, leading to degraded quality under large viewpoint shifts. For camera resimulation, generalizable reconstruction methods reduce reconstruction cost (Yang et al., 2025; Chen et al., 2024; Lu et al., 2025; Wang et al., 2025; Chen et al., 2026), while another line of work incorporates generative priors to repair artifacts in under-observed regions (Wu et al., 2025; Che et al., 2026; Schmid et al., 2026; Yin et al., 2025; Zhu et al., 2026; Wei et al., 2025). Both advances remain largely unexplored for LiDAR. LidarVerse combines these benefits for generalizable LiDAR resimulation: a single cross-scene generative model requires no per-scene optimization, completes content beyond the captured trajectory, and renders native spinning-LiDAR observations in real time.

3 Method

Fig. 2 provides an overview of LidarVerse. We learn a LiDAR-native tokenizer (Sec. 3.1), train a bidirectional teacher with a unified conditioning interface (Sec. 3.2), and distill it into a few-step causal student with persistent spatial memory (Sec. 3.3). Varying the conditions enables LiDAR generation, point-cloud forecasting, and generalizable resimulation with the same causal checkpoint (Sec. 3.4).

3.1 LiDAR-Native Tokenizer

3.1.1 Range-Image Representation

We represent each LiDAR sweep as a range image, a compact 2.5D representation compatible with existing video-generation infrastructure. A spinning LiDAR measures radial range along a set

of calibrated rays arranged into an $H \times W$ grid, where column i represents an azimuth angle θ_i and row j represents an elevation angle ϕ_j. The pixel value $r_{j,i}$ stores the radial range along the corresponding ray, producing a range image $\mathbf{R} \in \mathbb{R}^{H \times W}$. We clamp valid ranges to $[r_{\min}, r_{\max}]$ and normalize them to $[-1, 1]$. We additionally use a validity mask $\mathbf{M} \in \{0, 1\}^{H \times W}$ to distinguish valid measurements from missing returns. The default range representation concatenates $\mathbf{R}$ and $\mathbf{M}$ channel-wise, $\mathbf{I} = [\mathbf{R} \mid \mathbf{M}]$. When intensity is available, we append an intensity image as a third channel after clamping its values to $[0, 255]$ and normalizing them to $[0, 1]$.

3.1.2 LiDAR 4D Tokenizer

Given a T-frame range-image sequence $\mathbf{X} \in \mathbb{R}^{T \times H \times W \times C_{\text{in}}}$, where $C_{\text{in}} = 2$ for range and validity and $C_{\text{in}} = 3$ when intensity is included, we train an encoder $\mathcal{E}$ to map $\mathbf{X}$ to a latent $\mathbf{z} = \mathcal{E}(\mathbf{X}) \in \mathbb{R}^{\frac{T}{p_t} \times \frac{H}{p_s} \times \frac{W}{p_s} \times c}$, where p_s and p_t denote the spatial and temporal compression rates, respectively, and c is the latent dimension. A decoder $\mathcal{D}$ reconstructs the input as $\widehat{\mathbf{X}} = \mathcal{D}(\mathbf{z})$.

Spatial Neighborhood Attention RGB video tokenizers commonly use convolutional backbones (Team Wan et al., 2025). In LiDAR range images, however, neighboring angular bins may correspond to surfaces that are meters apart in 3D. Fixed convolutional aggregation can therefore blur depth discontinuities and produce the depth bleeding and wavy boundaries observed in prior work (Liu et al., 2026). We instead build our tokenizer from neighborhood attention (NATTEN) (Hassani et al., 2023). NATTEN performs content-adaptive aggregation within each local neighborhood while avoiding the quadratic memory cost of global self-attention.

Factorized Spatial and Temporal Attention Following latent video diffusion models (Blattmann et al., 2023), we incorporate temporal context by interleaving each spatial NATTEN block with temporal self-attention. We apply a causal temporal mask so that each frame depends only on the current and preceding frames, enabling compatibility with causal and variable-length rollouts.

Optimization We train the tokenizer with the following objective:

$$\mathcal{L}_{\text{tokenizer}} = \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{point}}\mathcal{L}_{\text{point}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}, \tag{1}$$

where $\mathcal{L}_{\text{depth}}$ is an ℓ_1 loss on valid reconstructed ranges, $\mathcal{L}_{\text{point}}$ is an ℓ_1 loss on valid 3D points derived from the reconstructed range image following Ran et al. (2024), $\mathcal{L}_{\text{mask}}$ is the binary cross-entropy loss on the predicted validity mask $\widehat{\mathbf{M}}$, and $\mathcal{L}_{\text{KL}}$ regularizes the latent posterior toward a standard normal prior. Latents are sampled using the reparameterization trick.

3.2 Bidirectional Teacher Model

We initialize our bidirectional teacher from Cosmos-Transfer2.5-2B (NVIDIA et al., 2025) and adapt the input and output projections to our LiDAR latent and conditioning channels. Given a clean LiDAR latent $\mathbf{z}$, we sample noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{\text{lat}})$ and a flow time $\tau \in [0, 1]$, where $\mathbf{I}_{\text{lat}}$ is the identity covariance in latent space, and construct $\mathbf{z}_\tau = (1 - \tau)\mathbf{z} + \tau\boldsymbol{\epsilon}, \mathbf{v}^* = \frac{\partial \mathbf{z}_\tau}{\partial \tau} = \boldsymbol{\epsilon} - \mathbf{z}$. We train the teacher to predict $\mathbf{v}^*$ under the unified conditioning interface described below.

3.2.1 Model Conditions

LiDAR Prefix The LiDAR prefix contains K observed sweeps preceding the target frames. We convert these sweeps to range images and encode them with the tokenizer. During training and sampling, their clean latents occupy the first K temporal positions and are concatenated temporally with the noisy target latents. A binary mask identifies the observed positions, and the generative loss is applied only to the target positions.

HD-Map and Actor Layout The layout condition contains lane centerlines, road markings, road boundaries, crosswalks, and 3D actor boxes. Given static map primitives $\mathcal{M}$, actor boxes $\mathcal{B}_t$, and target LiDAR-to-world pose $\mathbf{T}_t$, we transform these elements into the target LiDAR frame and rasterize them into a range-aligned representation. The resulting range image is encoded as $\mathbf{h}_t = \mathcal{E}(\mathbf{H}_t)$ and concatenated channel-wise with the corresponding noisy LiDAR latent.

Persistent Spatial Cache The PSC is a world-aligned geometric memory $\mathcal{G}$ obtained by accumulating point clouds from observations or previous generations. At a target sensor pose $\mathbf{T}_t$, we project the accumulated geometry into a range image, $\mathbf{G}_t = \Pi_{\text{geometry}}(\mathcal{G}; \mathbf{T}_t)$, where Π_{geometry} transforms world-frame points into the target LiDAR frame. We encode this image as $\mathbf{g}_t = \mathcal{E}(\mathbf{G}_t)$ and concatenate it channel-wise with the noisy LiDAR latent. During training, we construct a fixed PSC from ground-truth sweeps in two modes. The *static-background* mode removes points inside labeled dynamic-object boxes before accumulation. The *frozen-dynamic* mode additionally accumulates each actor in its local coordinate frame and places it according to its box pose at time t.

Condition Sampling Rather than training task-specific models, we sample all conditioning factors independently; see Appendix A.1 for details.

3.3 Causal Distillation

For real-time interactive simulation, we distill the teacher model into a few-step, block-autoregressive causal student and equip the student with complementary temporal and spatial memories for long-horizon rollout. The student retains the conditioning interface from Sec. 3.2, allowing efficiency to improve without changing how the simulation tasks are specified.

3.3.1 Self-Forcing Distillation

We initialize the causal student from the trained bidirectional teacher and replace full temporal attention with a block-causal pattern, where clean ground-truth latents from preceding blocks are stored in a K/V cache and each active block is independently perturbed and denoised. This stage adapts the teacher weights to causal attention before self-forcing distillation.

We then adopt self-forcing distribution matching distillation (SF-DMD) (Huang et al., 2025; Yin et al., 2024). Starting from the causal initialization, the causal student sequentially samples each four-frame block with four shifted rectified-flow SDE steps, commits the generated clean latents to its K/V cache, and updates the recurrent conditioning state before sampling the next block. Training therefore uses the same self-generated temporal and geometric histories encountered at inference. The conditioning factors remain independently randomized, so the distilled student retains the unified task interface learned by the teacher. Appendix A.2 gives the training objective.

3.3.2 Rolling Persistent Spatial Cache

A finite temporal K/V cache retains only recent latent context; increasing its length incurs growing memory and computational cost (Gu et al., 2025; Xi et al., 2026; Samuel et al., 2026). Once an observation leaves this window, the model may lose previously generated scene structure (Guo et al., 2026; Wu et al., 2026). We therefore maintain a sparse, world-aligned voxel map $\mathcal{V}$ as a rolling PSC, which uses the same PSC interface as in Sec. 3.2 but is constructed online from generated history.

After generating block b, we decode each latent $\widetilde{\mathbf{z}}_t$ into a point cloud $\widetilde{\mathcal{X}}_t$, register it to the common world frame using the corresponding sensor pose, and update the voxel map: $\mathcal{V}_b = \mathcal{U}(\mathcal{V}_{b-1}, \mathcal{Y}_b)$, where $\mathcal{Y}_b = \bigcup_{t \in \mathcal{T}_b} \mathbf{T}_t \widetilde{\mathcal{X}}_t$, and $\mathcal{T}_b$ contains the frame indices in block b and $\mathcal{U}$ is the cache update operator. Each occupied voxel v stores its most recently registered point $\mathbf{p}(v)$, an observation count $n^{\text{obs}}(v)$, and a removal count $n^{\text{rem}}(v)$. Each source frame contributes at most one observation to a voxel. When a voxel is observed, we update its point, increment its observation count, and reset its removal count. When valid free-space evidence contradicts a previously occupied voxel, we increment its removal count.

Before generating block $b+1$, we retain only confirmed, non-removed voxels and render them at each requested target pose:

$$\begin{aligned} \mathcal{P}(\mathcal{V}_b) &= \left\{ \mathbf{p}(v) \mid n^{\text{obs}}(v) \geq N_{\min},\ n^{\text{rem}}(v) < N_{\text{rem}} \right\}, \\ \mathbf{g}_t^{\text{cache}} &= \mathcal{E}(\Pi_{\text{geometry}}(\mathcal{P}(\mathcal{V}_b); \mathbf{T}_t)), \qquad t \in \mathcal{T}_{b+1}. \end{aligned} \tag{2}$$

Thus each block queries geometry produced only by earlier blocks. Its outputs update the cache only after generation, preventing self-conditioning and future leakage. During SF-DMD and inference, this rolling cache replaces the fixed ground-truth PSC without changing the conditioning channel or model architecture.

Table 1: **Tokenizer comparison.** ST means spatiotemporal. MMD values are multiplied by 10^5.

Method	ST Comp. rate	Reconstruction			Generation			TTCE ↓		CTC ↓		ICP Avg. ↓	TFLOPs /frame ↓
		CD ↓	L_1 ↓	AbsRel ↓	JSD ↓	MMD ↓	FUD ↓	3	4	3	4		
ST compression rate: 256x													
Cosmos CV NVIDIA et al. (2025)	8x8x4	0.439	0.790	0.031	0.061	6.41	46.26	1.773	2.561	0.322	**0.357**	0.089	**0.058**
Wan2.1 Team Wan et al. (2025)	8x8x4	0.297	0.395	0.017	**0.057**	**5.72**	37.52	1.983	2.832	0.337	0.381	0.077	0.911
Ours	16x16x1	**0.175**	**0.143**	**0.008**	0.061	6.87	**32.84**	**1.729**	**2.492**	**0.311**	**0.357**	**0.072**	0.400
ST compression rate: 32x / 64x													
LiDM Ran et al. (2024)	4x8x1	0.297	0.371	0.016	**0.057**	**5.48**	36.22	1.696	2.477	0.315	0.352	0.088	**0.212**
Cosmos LiDARGen Ren et al. (2025)	8x8x1	0.384	0.609	0.025	0.062	7.99	85.37	1.738	2.523	0.304	**0.338**	0.085	0.302
Ours	8x8x1	**0.115**	**0.072**	**0.005**	0.065	9.14	**18.17**	**1.681**	**2.463**	**0.294**	0.339	**0.059**	0.250

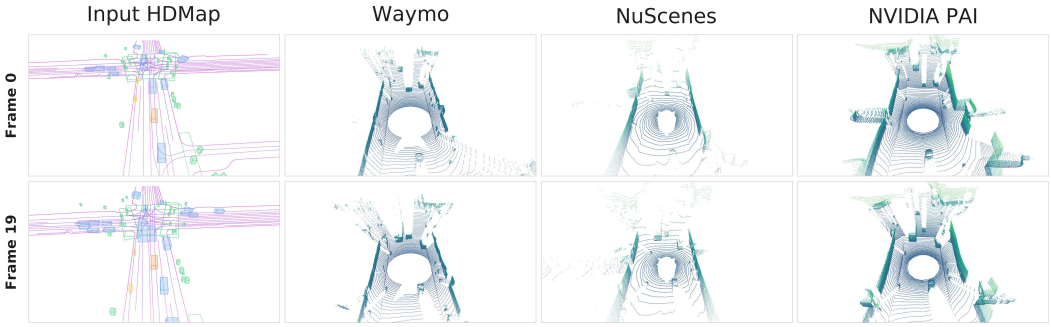

Figure 3: **Cross-sensor generation.** LidarVerse can generate diverse sensor patterns across datasets.

3.4 Model Inference

At inference, the conditions organize sensor simulation along a spectrum from new-world synthesis to reconstruction-grounded rendering. For **generation**, the model receives the HD-Map sequence and optionally a LiDAR prefix, and builds its rolling PSC from generated sweeps. For interactive action-conditioned generation, user-provided ego actions determine the future sensor poses. For **generalizable resimulation**, sparse observed sweeps are accumulated into a fixed frozen-dynamic PSC and rendered at novel target sensor poses. A LiDAR prefix can optionally provide temporal context; no HD-Map condition or per-scene optimization is required. Appendix A.5 details **forecasting** from observed history.

4 Experiments

We evaluate three questions: Does the tokenizer preserve metric geometry and support generative modeling? Can the model operate in real time across generation, forecasting, and resimulation? Which architectural and memory components drive performance? We primarily evaluate on Waymo (Sun et al., 2020), with extensions to intensity and multiple sensors and datasets. Appendices A.5 and A.8 report forecasting and intensity results; additional qualitative results follow.

4.1 Tokenizer Performance

Settings We study two compression regimes. At a matched $256\times$ spatiotemporal compression rate, we compare against the Cosmos CV (NVIDIA et al., 2025) and Wan2.1 (Team Wan et al., 2025) video tokenizers, both of which compress time by $4\times$. At lower compression rates of $32\times$ to $64\times$, we compare against the LiDM tokenizer (Ran et al., 2024) and the tokenizer used by Cosmos LiDARGen (Ren et al., 2025). To isolate the contribution of the latent representation, we pair every tokenizer with the same DiT architecture, training recipe, and evaluation protocol. For evaluation purposes, we train and evaluate each DiT on 9-frame setting.

Metrics We evaluate reconstruction on the Waymo validation split using Chamfer distance (CD), range L_1, and absolute relative error (AbsRel). For generation metrics, we measure distributional quality using BEV-based Maximum Mean Discrepancy (MMD) and Jensen–Shannon divergence (JSD), following Zyrianov et al. (2022). For perceptual metrics, we measure feature-space distribution quality using Fréchet Utonia Distance (FUD), computed with the universal point-cloud encoder Uto-

Table 2: **Autoregressive LiDAR generation.** We evaluate the causal LidarVerse checkpoint. MMD values are multiplied by 10^5. * indicates a reimplementation using our tokenizer and a matched DiT.

Method	Sampling steps	JSD ↓	MMD ↓	FUD ↓	TTCE ↓		CTC ↓				ICP Avg. ↓	FPS ↑
					3	4	1	2	3	4		
HD-Map-conditioned generation												
LiDARCrafter Liang et al. (2026)	256	0.065	14.13	97.24	**1.263**	**1.989**	0.278	0.360	0.441	0.519	0.120	0.127
LaGen* Zhou et al. (2025b)	35	0.075	19.19	50.29	1.584	2.296	**0.205**	0.268	0.330	0.390	0.071	0.402
Sensor2Sensor* Wang et al. (2026)	35	0.047	5.210	37.96	1.522	2.236	0.225	0.287	0.347	0.404	0.081	0.421
Ours	4	**0.046**	**4.390**	**25.01**	1.491	2.255	0.210	**0.253**	**0.293**	**0.331**	**0.068**	**11.18**
Interactive action-conditioned generation												
LiDARCrafter Liang et al. (2026)	256	0.137	54.383	70.38	**1.313**	**2.063**	0.258	0.335	0.410	0.483	0.104	0.135
LaGen* Zhou et al. (2025b)	35	0.064	12.51	42.31	1.913	2.621	0.256	0.362	0.460	0.550	0.074	0.754
Ours	4	**0.054**	**7.391**	**24.17**	1.517	2.280	**0.213**	**0.260**	**0.305**	**0.347**	**0.068**	**11.30**

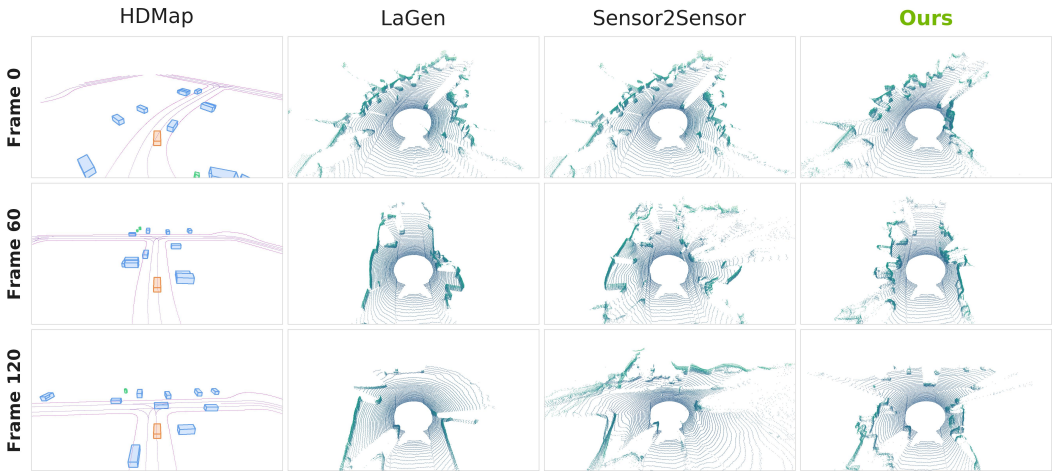

Figure 4: **HD-Map-Conditioned AR Generation.** Our results are realistic, layout-consistent, and stable under long-horizon rollouts.

nia (Zhang et al., 2026). We assess temporal consistency using Temporal Transformation Consistency Error (TTCE) and Chamfer Temporal Consistency (CTC) from Liang et al. (2026), together with ICP-Avg. from Zyrianov et al. (2025).

Results Tab. 1 shows that our LiDAR-native tokenizer preserves geometry substantially more faithfully in both compression regimes. Across both compression regimes, our tokenizers achieve better reconstruction and generative performance on most metrics when compared to other baselines. Most notably, we achieve the best perceptual and temporal consistency for generative results while remaining competitive on TFLOPs/frame.

Generalization Across Sensors and Datasets While most existing LiDAR tokenizers target a single sensor configuration, ours can accommodate multiple sensor configurations. We jointly fine-tune our tokenizer on 32-beam nuScenes (Caesar et al., 2020), 64-beam Waymo, and 128-beam NVIDIA PAI (NVIDIA, 2025), and fine-tune the bidirectional model on the same mixture. The bidirectional model can also generate distinct nuScenes, NVIDIA PAI, and Waymo LiDAR patterns from the same Waymo HD-Map input (Fig. 3).

4.2 Autoregressive LiDAR Generation

Settings We evaluate the causal student in two autoregressive LiDAR generation settings: HD-Map-conditioned generation and action-conditioned generation. In the first setting, the model produces 129-frame rollouts at 10 Hz from the corresponding HD-Map sequences. In the second, ego actions specify the future trajectory as described in Sec. 3.4. We use the same generative metrics as Sec. 4.1. To assess long-horizon stability (Fig. 5), we compute FUD over non-overlapping eight-frame windows (i.e., Windowed FUD), which balances reliable distributional estimates with sensitivity to rollout degradation.

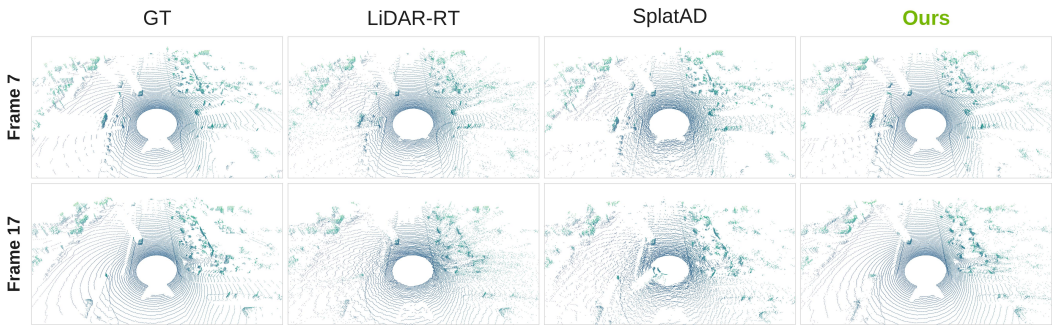

Figure 6: **Resimulation.** LidarVerse achieves high-quality reconstruction results in both interpolation (frame 7) and extrapolation (frame 17) settings while being generalizable (zero reconstruction time).

Table 3: **Generalizable LiDAR resimulation.** The causal student is our default real-time model; the bidirectional teacher is included as a quality reference. Per-scene optimization time is in minutes.

Method	All-202						Top-20 Travel Distance						Optim. time ↓	Real-time render
	Interpolation		Extrapolation		Aggregate		Interpolation		Extrapolation		Aggregate			
	L_1 ↓	CD↓	L_1 ↓	CD↓	L_1 ↓	CD↓	L_1 ↓	CD↓	L_1 ↓	CD↓	L_1 ↓	CD↓		
LiDAR4D Zheng et al. (2024)	1.129	0.188	1.881	0.460	1.364	0.273	1.999	0.998	3.284	1.769	2.401	1.239	89.8	×
LiDAR-RT Zhou et al. (2025a)	**1.089**	0.182	1.593	0.418	1.247	0.256	1.905	0.444	3.094	1.292	2.277	0.709	76.3	✓
SplatAD Hess et al. (2025)	1.799	0.265	2.325	0.408	1.964	0.309	3.125	0.357	4.654	0.947	3.603	0.542	15.0	✓
Ours (bidir)	1.124	**0.169**	**1.376**	**0.270**	**1.202**	**0.200**	**1.399**	**0.184**	**1.838**	**0.354**	**1.536**	**0.237**	**0.00**	×
Ours	1.279	0.235	1.682	0.397	1.405	0.285	1.593	0.260	2.215	0.370	1.787	0.294	**0.00**	✓

HD-Map-Conditioned Generation Results We report our results in Tab. 2. LidarVerse outperforms all baselines on the statistical and perceptual distribution metrics. It remains competitive on short-term temporal consistency and performs best over longer temporal offsets, indicating that causal distillation preserves the teacher's generative quality without sacrificing long-horizon coherence. The qualitative results in Fig. 4 likewise remain structurally stable as the baselines visibly degrade.

Action-Conditioned Generation Results The bottom half of Tab. 2 evaluates interactive action-conditioned generation. Given the first ground-truth frame, each method generates 128 frames along the ground-truth ego trajectory. We omit Sensor2Sensor because it has no native mechanism for injecting ego poses. LidarVerse noticeably outperforms the remaining methods across the reported distributional, perceptual metrics, and temporal metrics. The windowed FUD curves in Fig. 5 further show that its quality degrades much more slowly over time, demonstrating LidarVerse's robustness and stability during long-horizon rollouts.

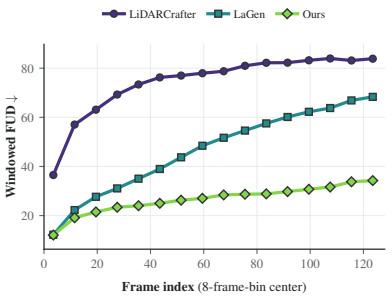

Figure 5: **Long-horizon AR rollouts.**

4.3 Generalizable LiDAR Resimulation

Settings We report results on both the standard validation split and its top-20 furthest travel-distance scenes to better understand performance in fast moving scenes. Appendix A.3 lists the exact log IDs. Following Yang et al. (2025) and Chen et al. (2026), we split each validation log into non-overlapping 20-frame chunks, use frames 0, 4, 9, and 14 as input, and evaluate on the remaining frames. We report results in three settings: interpolation (frames 0–14 excluding the inputs), extrapolation (frames 15–19), and aggregate (frames 0–19 excluding the inputs).

Results Tab. 3 shows that despite requiring no test-time optimization, our causal student achieves competitive performance with the per-scene reconstruction baselines and outperforms them in the fast-moving setting. The bidirectional teacher further shows the best fidelity across nearly all reported metrics. The qualitative results in Fig. 6 further show that the causal checkpoint preserves captured scene structure in both interpolation and extrapolation views. To our knowledge, LidarVerse is the first real-time generalizable LiDAR resimulation model of this capability.

Table 4: **Tokenizer architecture ablation on the Waymo dataset.** We report representative reconstruction, generation, temporal-consistency, and efficiency metrics, demonstrating the superior performance of our attention backbone compared with the conventional convolutional backbone.

Backbone	Recon.			Generation			TFLOPs /frame↓
	L_1 ↓	CD ↓	AbsRel ↓	FUD ↓	CTC@4 ↓	TTCE@4 ↓	
Conv.	0.297	0.278	0.015	26.54	0.341	2.496	0.575
Attn.	**0.072**	**0.115**	**0.005**	**18.17**	**0.339**	**2.463**	**0.250**

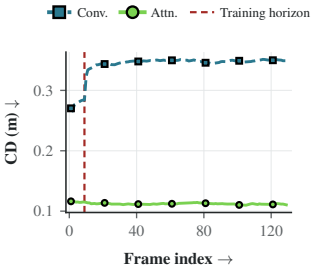

Figure 7: **Long-sequence recon.**

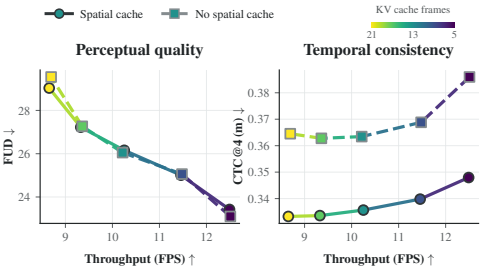

Figure 8: **Spatial and K/V cache size ablation.**

Table 5: **Unified vs. Specialist comparison.** MMD values are multiplied by 10^5. Resimulation metrics are aggregated over the top-20 travel distance subset.

Method	Generation			Resimulation	
	FUD ↓	MMD ↓	ICP Avg. ↓	L_1 ↓	CD ↓
Specialist	**14.67**	6.950	0.047	**1.380**	**0.235**
Unified	16.05	**6.843**	**0.044**	1.536	0.237

4.4 Ablation

LiDAR-Native Tokenizer Architecture We isolate the effect of spatial neighborhood attention by comparing our tokenizer against a convolutional tokenizer at the same compression rate and latent dimensionality. As shown in Tab. 4, neighborhood attention achieves better reconstruction on all metrics and improves long-horizon temporal consistency while reducing tokenizer compute by 57%. Fig. 7 further shows that attention enables more stable reconstruction beyond the training horizon.

Spatial Cache and Temporal K/V Cache We jointly vary the temporal K/V cache length over $\{5, 9, 13, 17, 21\}$ frames and either enable the PSC condition or remove it. Fig. 8 shows the resulting quality–throughput trade-off. Removing the spatial cache substantially worsens temporal consistency while having little effect on FUD. Increasing the K/V cache size improves temporal consistency but reduces throughput and fidelity. We use a nine-frame K/V cache because it preserves real-time generation while providing a substantial consistency gain over shorter temporal contexts.

Unified versus Task-Specific Specialists We compare our unified bidirectional model with two task-specific specialists for generation and resimulation. Each specialist is trained with the same budget as the unified model and evaluated using the corresponding protocol in Secs. 4.2 (using 21 frames instead of 129 frames) and 4.3. The results in Tab. 5 demonstrate that unified training retains competitive performance across both paradigms with a single checkpoint.

5 Conclusion

We presented LidarVerse, a real-time LiDAR world model unifying generative and reconstruction-grounded sensor simulation. Within one conditional framework, scene-level conditions guide new-world generation, a LiDAR prefix enables forecasting, and a PSC anchors resimulation to captured geometry without per-scene optimization. A universal tokenizer provides a shared latent space across sensors, and a bidirectional teacher learns the common conditioning interface. Distillation yields a four-step causal student with a temporal K/V cache for recent dynamics and a world-aligned spatial cache for persistent geometry. This checkpoint operates at or above the native 10-Hz LiDAR rate and achieves state-of-the-art performance on most reported metrics across generation, forecasting, and resimulation. Together, these results show that changing the available scene evidence, rather than the model architecture, can unify generative and reconstruction-based simulation within one model.

6 AI USE STATEMENT

In this work, we used generative AI tools for **"implement methods", "design or provide feedback on research methodology or experiments", "support qualitative and thematic data analysis", "interpret results"**. We have not used generative AI tools for **"generate synthetic datasets", "clean and reformat dataset"** and **"assist with translation", "help develop theoretical models or conceptual frameworks", "formulate mathematical claims", "provide critical ingredients for proving mathematical claims", "assist in the writing of proofs", "propose or refine hypotheses"** are not applicable to this work. Additionally, we used generative AI tools for **"create or edit software code", "edit a research paper to improve readability", "propose a title or keywords for a research paper"** . We have reviewed all AI-assisted work. We have proofread all materials generated by AI, and manually check the citation one by one. We also have carefully checked the code generated, and confirm the reproducibility across different environments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Ben Agro, Quinlan Sykora, Sergio Casas, Thomas Gilles, and Raquel Urtasun. UnO: Unsupervised occupancy fields for perception and forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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 (CVPR)*, 2020.

Henry Che, Jingkang Wang, Yun Chen, Ze Yang, Sivabalan Manivasagam, and Raquel Urtasun. Diffusion-guided generalizable enhancer for urban scene reconstruction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Xiaoxue Chen, Ziyi Xiong, Yuantao Chen, Gen Li, Nan Wang, Hongcheng Luo, Long Chen, Haiyang Sun, Bing Wang, Guang Chen, Hongyang Li, Ya-Qin Zhang, Hangjun Ye, and Hao Zhao. DGGT: Feedforward 4D reconstruction of dynamic driving scenes using unposed images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Yun Chen, Jingkang Wang, Ze Yang, Sivabalan Manivasagam, and Raquel Urtasun. G3R: Gradient guided generalizable reconstruction. In *European Conference on Computer Vision (ECCV)*, 2024.

Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction. *arXiv preprint arXiv:2503.19325*, 2025.

Yanjun Guo, Zhengqiang Zhang, Pengfei Wang, Xinyue Liang, Zhiyuan Ma, and Lei Zhang. Memorize when needed: Decoupled memory control for spatially consistent long-horizon video generation. *arXiv preprint arXiv:2604.18215*, 2026.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Georg Hess, Carl Lindström, Maryam Fatemi, Christoffer Petersson, and Lennart Svensson. SplatAD: Real-time lidar and camera rendering with 3D gaussian splatting for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Qianjiang Hu, Zhimin Zhang, and Wei Hu. RangeLDM: Fast realistic LiDAR point cloud generation. *arXiv preprint arXiv:2403.10094*, 2024.

Shengyu Huang, Zan Gojcic, Zian Wang, Francis Williams, Yoni Kasten, Sanja Fidler, Konrad Schindler, and Or Litany. Neural LiDAR fields for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. Point cloud forecasting as a proxy for 4D occupancy forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, Shuchang Zhou, Li Zhang, Xiaojuan Qi, Hao Zhao, Mu Yang, Wenjun Zeng, and Xin Jin. UniScene: Unified occupancy-centric driving scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Ao Liang, Youquan Liu, Yu Yang, Dongyue Lu, Linfeng Li, Lingdong Kong, Huaici Zhao, and Wei Tsang Ooi. LiDARCrafter: Dynamic 4D world modeling from LiDAR sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

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 (CVPR)*, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Hao Lu, Tianshuo Xu, Wenzhao Zheng, Yunpeng Zhang, Wei Zhan, Dalong Du, Masayoshi Tomizuka, Kurt Keutzer, and Yingcong Chen. DrivingRecon: Large 4D gaussian reconstruction model for autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

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 (ICRA)*, 2025.

NVIDIA. NVIDIA Physical AI Autonomous Vehicles Dataset. `https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles`, 2025.

NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue

Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

NVIDIA, Alisson Azzolini, Hannah Brandon, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, Francesco Ferroni, Rama Govindaraju, Jinwei Gu, Siddharth Gururani, Imad El Hanafi, Zekun Hao, Jacob Huffman, Jingyi Jin, Brendan Johnson, Rizwan Khan, George Kurian, Elena Lantz, Nayeon Lee, Zhaoshuo Li, Xuan Li, Tsung-Yi Lin, Yen-Chen Lin, Ming-Yu Liu, Andrew Mathau, Yun Ni, Lindsey Pavao, Wei Ping, David W. Romero, Misha Smelyanskiy, Shuran Song, Lyne Tchapmi, Andrew Z. Wang, Boxin Wang, Haoxiang Wang, Fangyin Wei, Jiashu Xu, Yao Xu, Xiaodong Yang, Zhuolin Yang, Xiaohui Zeng, and Zhe Zhang. Cosmos-Reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

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 (CVPR)*, 2024.

Xuanchi Ren, Yifan Lu, Hanxue Liang, Jay Zhangjie Wu, Huan Ling, Mike Chen, Sanja Fidler, Francis Williams, and Jiahui Huang. SCube: Instant large-scale scene reconstruction using VoxSplats. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Xuanchi Ren, Yifan Lu, Tianshi Cao, Ruiyuan Gao, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, Seung Wook Kim, Jun Gao, Laura Leal-Taixé, Mike Chen, Sanja Fidler, and Huan Ling. Cosmos-Drive-Dreams: Scalable synthetic driving data generation with world foundation models. arXiv preprint arXiv:2506.09042, 2025.

Dvir Samuel, Issar Tzachor, Matan Levy, Michael Green, Gal Chechik, and Rami Ben-Ari. Fast autoregressive video diffusion and world models with temporal cache compression and sparse attention. *arXiv preprint arXiv:2602.01801*, 2026.

Katharina Schmid, Nicolas von Lützow, Jozef Hladký, Angela Dai, and Matthias Nießner. GenRecon: Bridging generative priors for multi-view 3D scene reconstruction. *arXiv preprint arXiv:2605.23888*, 2026.

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

Tang Tao, Longfei Gao, Guangrun Wang, Yixing Lao, Peng Chen, Hengshuang Zhao, Dayang Hao, Xiaodan Liang, Mathieu Salzmann, and Kaicheng Yu. LiDAR-NeRF: Novel LiDAR view synthesis via neural radiance fields. *arXiv preprint arXiv:2304.10406*, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. NeuRAD: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Haithem Turki, Qi Wu, Xin Kang, Janick Martinez Esturo, Shengyu Huang, Ruilong Li, Zan Gojcic, and Riccardo de Lutio. SimULi: Real-time LiDAR and camera simulation with unscented transforms. In *International Conference on Learning Representations (ICLR)*, 2026.

Jiahao Wang, Bo Sun, Yijing Bai, Vincent Casser, Songyou Peng, Zehao Zhu, Meng-Li Shih, Xander Masotto, Shih-Yang Su, Kanaad V. 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 (CVPR)*, 2026.

Jingkang Wang, Henry Che, Yun Chen, Ze Yang, Lily Goli, Sivabalan Manivasagam, and Raquel Urtasun. Flux4D: Flow-based unsupervised 4D reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jiaxin Wei, Stefan Leutenegger, and Simon Schaefer. GSFix3D: Diffusion-guided repair of novel views in gaussian splatting. *arXiv preprint arXiv:2508.14717*, 2025.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

Hanfeng Wu, Xingxing Zuo, Stefan Leutenegger, Or Litany, Konrad Schindler, and Shengyu Huang. Dynamic LiDAR re-simulation using compositional neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. DIFIX3D+: Improving 3D reconstructions with single-step diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Xindi Wu, Sven Elflein, James Lucas, Olga Russakovsky, Laura Leal-Taixé, Despoina Paschalidou, Jonathan Lorraine, and Aljoša Ošep. Addressable memory for video world models. *arXiv preprint arXiv:2608.07408*, 2026.

Haocheng Xi, Shuo Yang, Yilong Zhao, Muyang Li, Han Cai, Xingyang Li, Yujun Lin, Zhuoyang Zhang, Jintao Zhang, Xiuyu Li, Zhiying Xu, Jun Wu, Chenfeng Xu, Ion Stoica, Song Han, and Kurt Keutzer. Quant VideoGen: Auto-regressive long video generation via 2-bit KV-cache quantization. *arXiv preprint arXiv:2602.02958*, 2026.

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 *IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.

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 (CVPR)*, 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 (CVPR)*, 2026.

Jiawei Yang, Jiahui Huang, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Maximilian Igl, Apoorva Sharma, Peter Karkus, Danfei Xu, Boris Ivanovic, Yue Wang, and Marco Pavone. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. *arXiv preprint arXiv:2501.00602*, 2025.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. UniSim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Xingyilang Yin, Qi Zhang, Jiahao Chang, Ying Feng, Qingnan Fan, Xi Yang, Chi-Man Pun, Huaqi Zhang, and Xiaodong Cun. GSFixer: Improving 3D gaussian splatting with reference-guided video diffusion priors. *arXiv preprint arXiv:2508.09667*, 2025.

Yujia Zhang, Xiaoyang Wu, Yunhan Yang, Xianzhe Fan, Han Li, Yuechen Zhang, Zehao Huang, Naiyan Wang, and Hengshuang Zhao. Utonia: Toward one encoder for all point clouds. In *International Conference on Machine Learning (ICML)*, 2026.

Zehan Zheng, Fan Lu, Weiyi Xue, Guang Chen, and Changjun Jiang. LiDAR4D: Dynamic neural fields for novel space-time view LiDAR synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Chenxu Zhou, Lvchang Fu, Sida Peng, Yunzhi Yan, Zhanhua Zhang, Yong Chen, Jiazhi Xia, and Xiaowei Zhou. LiDAR-RT: Gaussian-based ray tracing for dynamic LiDAR re-simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Sizhuo Zhou, Xiaosong Jia, Fanrui Zhang, Junjie Li, Juyong Zhang, Yukang Feng, Jianwen Sun, Songbur Wong, Junqi You, and Junchi Yan. LaGen: Towards autoregressive LiDAR scene generation. *arXiv preprint arXiv:2511.21256*, 2025b.

Liyuan Zhu, Manjunath Narayana, Michal Stary, Will Hutchcroft, Gordon Wetzstein, and Iro Armeni. GaussFusion: Improving 3D reconstruction in the wild with a geometry-informed video generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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

Vlas Zyrianov, Henry Che, Zhijian Liu, and Shenlong Wang. LidarDM: Generative LiDAR simulation in a generated world. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

A Appendix

A.1 Model Details

Tokenizer Our tokenizer is a symmetric Transformer with channel widths $[128, 256, 512]$, spatial depths $[4, 2, 4]$, and $[4, 16, 16]$ attention heads, corresponding to ten spatial Transformer blocks in each of the encoder and decoder. The spatial NATTEN (Hassani et al., 2023) blocks use a 5×35 neighborhood window with no dilation and circular padding, following Nakashima et al. (2025), to handle azimuthal discontinuities. Factorized temporal-attention blocks are applied at the two lower-resolution stages, while no temporal downsampling is performed. The tokenizer uses a 2×2 input patch embedding, an MLP expansion ratio of 3, two latent-mapping layers, and a 64-channel latent space. It compresses each sweep spatially by 8×8, mapping an input of size $2 \times 64 \times 1024$ to a latent of size $64 \times 8 \times 128$ while preserving temporal resolution. The decoder predicts both the range image and its validity mask. We train the tokenizer for $120\,000$ iterations. The total parameter count for the tokenizer is 64.5M. We set $\lambda_{\text{depth}} = \lambda_{\text{point}} = \lambda_{\text{mask}} = 1$ and $\lambda_{\text{KL}} = 10^{-6}$. For our final tokenizer, we use $p_s = 8$, $p_t = 1$, and $c = 64$.

Bidirectional and Causal Model Training Our DiT is initialized from Cosmos-Transfer2.5-2B (NVIDIA et al., 2025) and subsequently adapted to the LiDAR latent space. It contains 28 blocks with a hidden width of 2048 and 16 attention heads. Each block applies spatiotemporal self-attention, text cross-attention when appropriate, and an MLP with AdaLN-LoRA time conditioning. We retain the 3D rotary positional embeddings and flow-matching formulation (Lipman et al., 2023; Liu et al., 2023). We train the world model in three stages using 21-frame clips. *First*, we train the bidirectional flow-matching teacher for $270\,000$ iterations. We first pretrain for HDMap-conditioned generation only with 9-frames for $150\,000$, then 21-frames generation for $100\,000$ iterations before finetuning for unified conditioning for additional $20\,000$ iterations. Empirically, we found that this curriculum-style training produced better results than directly using the joint conditioning. *Second*, we initialize the causal student from the teacher and train it with teacher forcing for $10\,000$ iterations, using one prefix frame followed by causal blocks of four frames. *Finally*, we perform self-forcing distribution matching on the causal student for $2\,000$ iterations. We update the fake-score network every cycle and the student every five cycles. The final causal student generates each four-frame block using four steps and maintains a rolling nine-frame K/V cache including one persistent sink frame. The four-step SDE noise schedule is $(1, 15/16, 5/6, 5/8, 0)$, with zero denoting the clean endpoint. We provide detailed training configurations in Tab. 6.

Condition Sampling Rather than training task-specific models, we sample all conditioning factors independently. With probability 0.5, we sample the *LiDAR prefix* length K uniformly from $\{1, 2, 4, 8, 10, 12, 16\}$; otherwise, $K = 0$. We drop the *HD-Map* condition with probability 0.5 using a null token. For the *PSC*, we sample null geometry, static-background geometry, and frozen-dynamic geometry with probabilities 0.1, 0.45, and 0.45, respectively. In the frozen-dynamic mode, the number of source sweeps $N_{\mathcal{G}}$ is sampled uniformly from $\{1, 2, 4, 8, 10, 12, 16, 20\}$. For the text-conditioned results in Fig. 12, we use 20% classifier-free dropout of the text embeddings. This factorized sampling exposes the model to diverse combinations of scene evidence without task-specific tokens or objectives, producing the shared conditional prior used for causal distillation.

Cross-Sensor Conditioning and Checkpoints For the cross-sensor experiments, we jointly fine-tune the Waymo-trained tokenizer on 32-beam nuScenes, 64-beam Waymo, and 128-beam NVIDIA PAI. We then fine-tune the bidirectional teacher on the same mixture, using an additional sensor embedding that identifies each dataset's LiDAR configuration while retaining the architecture and training recipe. The intensity extension follows the same procedure with a tokenizer and teacher trained on both range and intensity. Unless explicitly identified as bidirectional, the world-model results use the causal student; the cross-sensor and intensity extensions described here use the fine-tuned bidirectional teacher.

A.2 Additional Details on Causal Model

Self-Forcing Distillation Objective Rather than regressing the rollout to a paired ground-truth sequence, SF-DMD matches its distribution to the real LiDAR latent distribution. Let $\widetilde{\mathbf{z}}$ denote a

Table 6: **Training configurations for our three-stage pipeline.**

Configuration	Bidirectional training	Causal initialization	Self forcing distillation
Global batch size	32	32	32
Training clip length	$9 \rightarrow 21$ frames	21 frames	21 frames
Model optimizer	AdamW $\epsilon = 10^{-8}$ $\mathrm{lr} = 10^{-5}$ $\beta = (0.9, 0.999)$ $\mathrm{wd} = 0.01$	AdamW $\epsilon = 10^{-8}$ $\mathrm{lr} = 10^{-5}$ $\beta = (0.9, 0.999)$ $\mathrm{wd} = 0.01$	AdamW $\epsilon = 10^{-8}$ $\mathrm{lr} = 2 \times 10^{-6}$ $\beta = (0.9, 0.999)$ $\mathrm{wd} = 0.01$
Fake-score optimizer	–	–	AdamW $\epsilon = 10^{-8}$ $\mathrm{lr} = 4 \times 10^{-7}$ $\beta = (0.9, 0.999)$ $\mathrm{wd} = 0.01$
Learning-rate schedule	Constant	Constant	Constant
RF timestep shift	5	5	5
Distillation teacher CFG	–	–	3.0
Student sampling	–	–	Four-step SDE, $\eta = 1$
DiT precision	BF16	BF16	BF16
Training iterations	270,000	10,000	2,000

self-rolled-out latent sequence. After perturbing it to $\widetilde{\mathbf{z}}_\tau$, a frozen real-data denoiser and a learned generated-data denoiser produce clean-latent estimates $\widehat{\mathbf{z}}_0^{\mathrm{real}}$ and $\widehat{\mathbf{z}}_0^{\mathrm{gen}}$, respectively. For a replayed denoising step, define the adaptive scale $a = \mathrm{mean}\,|\widetilde{\mathbf{z}} - \widehat{\mathbf{z}}_0^{\mathrm{real}}| + \varepsilon$. The student's clean prediction $\widehat{\mathbf{z}}_0^{\mathrm{stu}}$ is optimized with

$$\mathcal{L}_{\text{SF-DMD}} = \mathbb{E}\left[\frac{1}{2}\left\|\widehat{\mathbf{z}}_0^{\mathrm{stu}} - \mathrm{sg}\left[\widetilde{\mathbf{z}} - \frac{\widehat{\mathbf{z}}_0^{\mathrm{gen}} - \widehat{\mathbf{z}}_0^{\mathrm{real}}}{a}\right]\right\|_2^2\right]. \tag{3}$$

Here, sg denotes stop-gradient and a adaptively normalizes the update. The real-data denoiser remains frozen, while the generated-data denoiser is initialized from it and trained with the rectified-flow objective on self-rollouts. The conditioning factors remain independently randomized, preserving the teacher's unified task interface.

Rolling Spatial Cache We use $0.15\,\mathrm{m}$ cubic voxels, each storing its latest point, observation count, and removal count. Each generated frame increments the observation count at most once per voxel. In case of repeated observation within a voxel, we retain the latest point and reset the removal count. To detect stale geometry, we project eligible cached points into the last sweep's range image. If the surrounding 3×3 pixels contain valid returns and are farther from the cached point by more than $0.75\,\mathrm{m}$, we increment its voxel's removal count.

For conditioning, only voxels with at least $N_{\min} = 3$ observations contribute to the rendered PSC. Voxels reaching $N_{\mathrm{rem}} = 2$ contradictions are deleted together with their counts. If those voxels are observed again, they are registered anew and must satisfy the observation threshold again. Cache updates occur only after block generation, preventing a block from conditioning on its own outputs.

A.3 Dataset & Evaluation Details

Datasets We use Waymo Open Dataset (Sun et al., 2020) for model training and evaluation. Its training split has 798 sequences and validation split has 202 sequences. Each sequence contains approximately 20 seconds of LiDAR captured at 10 Hz, giving roughly 160K training sweeps, 4.4 hours of sensor data, and 142.8K possible overlapping 21-frame training clips. Both the tokenizer and all three stages of world-model training use this split. For the cross-sensor experiments in Sec. 4.1, we additionally fine-tune the tokenizer and DiT on NVIDIA PAI (NVIDIA, 2025), Waymo, and nuScenes (Caesar et al., 2020). This mixture contains approximately 640K unique 21-frame clips from 111,988 NVIDIA PAI sequences, 798 Waymo sequences, and 722 nuScenes sequences. To prevent the substantially larger NVIDIA PAI corpus from dominating training, we sample the three datasets with equal probability.

Table 7: **Top-20 Travel Distance Sequence ID** Sequences are ranked by longest traveled distance.

Rank	Sequence ID	Rank	Sequence ID
1	2506799708748258165_6455_000_6475_000	11	16767575238225610271_5185_000_5205_000
2	5372281728627437618_2005_000_2025_000	12	13573359675885893802_1985_970_2005_970
3	6491418762940479413_6520_000_6540_000	13	2713381581363292802541_070_2561_070
4	15396462829361334065_4265_000_4285_000	14	10203656353524179475_7625_000_7645_000
5	18305329035161925340_4466_730_4486_730	15	10335539493577748957_1372_870_1392_870
6	13415985003725220451_6163_000_6183_000	16	1457696187335927618_595_027_615_027
7	9164052963393400298_4692_970_4712_970	17	2105808889850693535_2295_720_2315_720
8	10837554759555844344_6525_000_6545_000	18	11048712972908676520_545_000_565_000
9	4490196167747784364_616_569_636_569	19	14165166478774180053_1786_000_1806_000
10	15096340672898807711_3765_000_3785_000	20	7163140554846378423_2717_820_2737_820

Top-20 Travel Distance Validation Subset. We rank all 202 Waymo validation sequences by total ego path length, computed as the sum of 3D translation distances between consecutive sweep-start LiDAR poses over the full sequence. We retain the 20 longest paths and divide each into nine non-overlapping 20-frame windows, yielding 180 windows. We report the sequence ID in Tab. 7.

LiDAR Generation Baseline Implementation For LiDARCrafter (Liang et al., 2026), we use the released codebase and train from scratch following our protocol. For LaGen (Zhou et al., 2025b) and Sensor2Sensor (Wang et al., 2026) whose code was not publicly available when these experiments were conducted, we reimplement the methods using our tokenizer, a matched DiT backbone, and the same training protocol. We adapt the methods to HD-Map conditioning and action conditioning where needed. These reimplemented results are marked with * in Tab. 2.

Ablation Evaluation Details For the long-sequence tokenizer evaluation in Fig. 7, both backbones are trained on nine-frame sequences and evaluated on 129-frame sequences from 202 Waymo validation clips using EMA weights. The system ablations use the 129-frame causal-generation protocol from Sec. 4.2. In Fig. 8, perceptual and temporal metrics use all 202 Waymo validation logs; throughput is steady-state FPS over ten matched clips on an H100 GPU. We compare the PSC condition against the learned null-geometry condition while varying the temporal K/V cache length. Figure 19 uses the same 202 logs and a fixed nine-frame K/V cache; the ring marks the selected $N_{\min} = 3$ operating point used for the main causal-generation results.

A.4 Bidirectional Teacher Model Generation Results

Before evaluating real-time autoregressive inference, we test whether the bidirectional teacher has learned a useful shared conditional prior. Because it attends to the full sequence during denoising, it isolates the representational capacity and conditioning flexibility of the base model from errors introduced by causal rollout. All results in this subsection use the teacher checkpoint from Sec. 3.2. Together, the results below show that one bidirectional prior can synthesize diverse scenes while responding to temporal, geometric, and semantic evidence through a common interface.

Unconditional Generation We present unconditional generation results from our bidirectional teacher in Fig. 9. The model generates realistic and diverse 4D LiDAR sequences from noise, ranging from high-speed highway scenarios (Sample 0) to dense urban traffic (Sample 1).

HD-Map-Conditioned Generation To assess condition alignment, we present HD-Map-conditioned generation results from our teacher in Fig. 10. The model generates 21 frames conditioned on corresponding 21-frame HD-Map sequences. It follows the conditioning well across highway (Sample 0), pedestrian-rich (Sample 1), and urban (Sample 2) scenarios.

First-Frame-Conditioned Generation We further show first-frame-conditioned generation in Fig. 11. Given one observed LiDAR frame, the model generates the next 21 frames. The resulting sequences contain plausible future motion in which agents follow the traffic structure.

Text-Controlled Weather Generation Finally, we show text-controlled weather generation on the NVIDIA PAI dataset in Fig. 12. We use Qwen2.5-VL-7B-Instruct Bai et al. (2025) to produce detailed weather descriptions from front-view camera images and Cosmos-Reason1 NVIDIA et al. (2025)

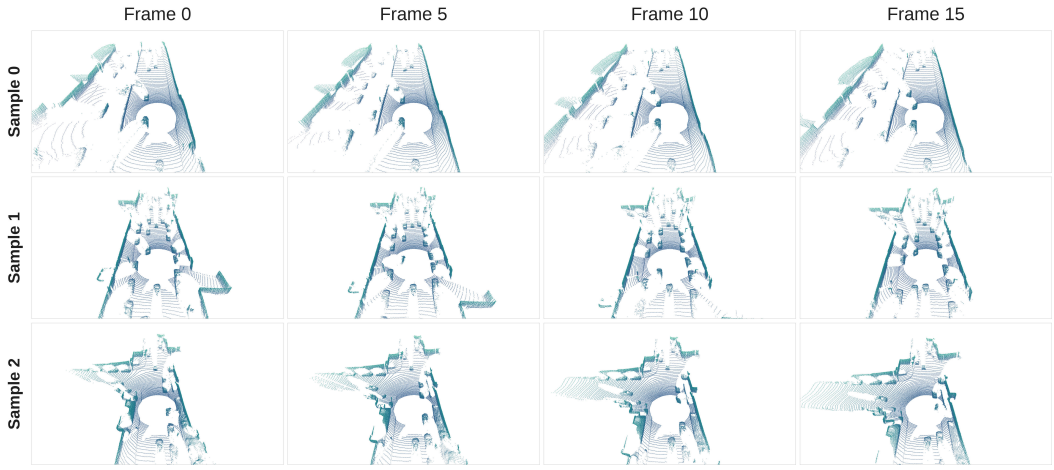

Figure 9: **Unconditional generation with the bidirectional model.** LidarVerse generates high-quality, diverse sequences. Sample 0 depicts a highway scene, Sample 1 dense traffic, and Sample 2 an intersection. All conditioning inputs are set to null.

Table 8: **Point-cloud forecasting.** * indicates reimplementation to the best of our ability.

Method	1 s			2 s			3 s			Aggregated		
	$L_1 \downarrow$	AbsRel ↓	CD ↓	$L_1 \downarrow$	AbsRel ↓	CD ↓	$L_1 \downarrow$	AbsRel ↓	CD ↓	$L_1 \downarrow$	AbsRel ↓	CD ↓
Naive raycasting (Khurana et al., 2023)	2.461	0.171	1.584	3.161	0.217	3.343	3.768	0.251	6.049	2.807	0.193	2.739
4D-Occ (Khurana et al., 2023)	2.146	0.113	2.173	2.602	0.139	3.230	3.052	0.170	4.575	2.369	0.126	2.924
UnO* (Agro et al., 2024)	2.040	0.112	1.000	2.300	0.130	1.117	2.654	0.165	1.440	2.208	0.126	1.070
Ours	**1.482**	**0.090**	**0.485**	**2.104**	**0.126**	**0.805**	**2.554**	**0.156**	**1.328**	**1.768**	**0.108**	**0.676**

to encode the text into embeddings and condition the bidirectional model through cross-attention at every DiT layer. The model reproduces typical rainy-weather LiDAR patterns, including increased ray drop on the road surface.

A.5 Point-Cloud Forecasting

Forecasting introduces more observed evidence than generation: a LiDAR prefix specifies the recent scene state and dynamics, but the future remains unobserved. We evaluate whether the same causal generator can use this prefix to predict future measurements without a task-specific forecasting head.

Inference The observed LiDAR history supplies the prefix and initializes the temporal K/V cache and PSC. Without an HD-Map condition, the same causal checkpoint predicts future sweeps at the provided ego poses, updating both caches from generated history. Forecasting therefore changes the available evidence and cache context rather than the model weights or prediction head.

Settings Following the point-cloud forecasting protocol of Khurana et al. (2023), the model predicts a three-second future horizon from three seconds of point-cloud history. We use data sampled at 10 Hz, resulting in 30 history frames and 30 future frames. We compare against naive raycasting, 4D-Occ Khurana et al. (2023), and the state-of-the-art UnO method Agro et al. (2024). Because UnO has not released its code publicly, we reimplement it to the best of our ability based on the details in the paper. For our method, we use a K/V cache size of 21 rather than 9, as we empirically find that it performs better for forecasting. We report L_1 error, absolute relative error (AbsRel), and inner Chamfer distance (CD) at the one-, two-, and three-second marks. The *Aggregated* column reports the mean of each metric across all 30 predicted frames.

Results Tab. 8 shows that LidarVerse outperforms the task-specific baselines on most metrics across all forecast horizons. The visualizations in Fig. 14 also show plausible future motion while preserving the shapes of traffic participants. Naive raycasting struggles when the future viewpoint reveals regions absent from the input history, as in the freeway example; the shared generative prior can instead complete plausible scene structure.

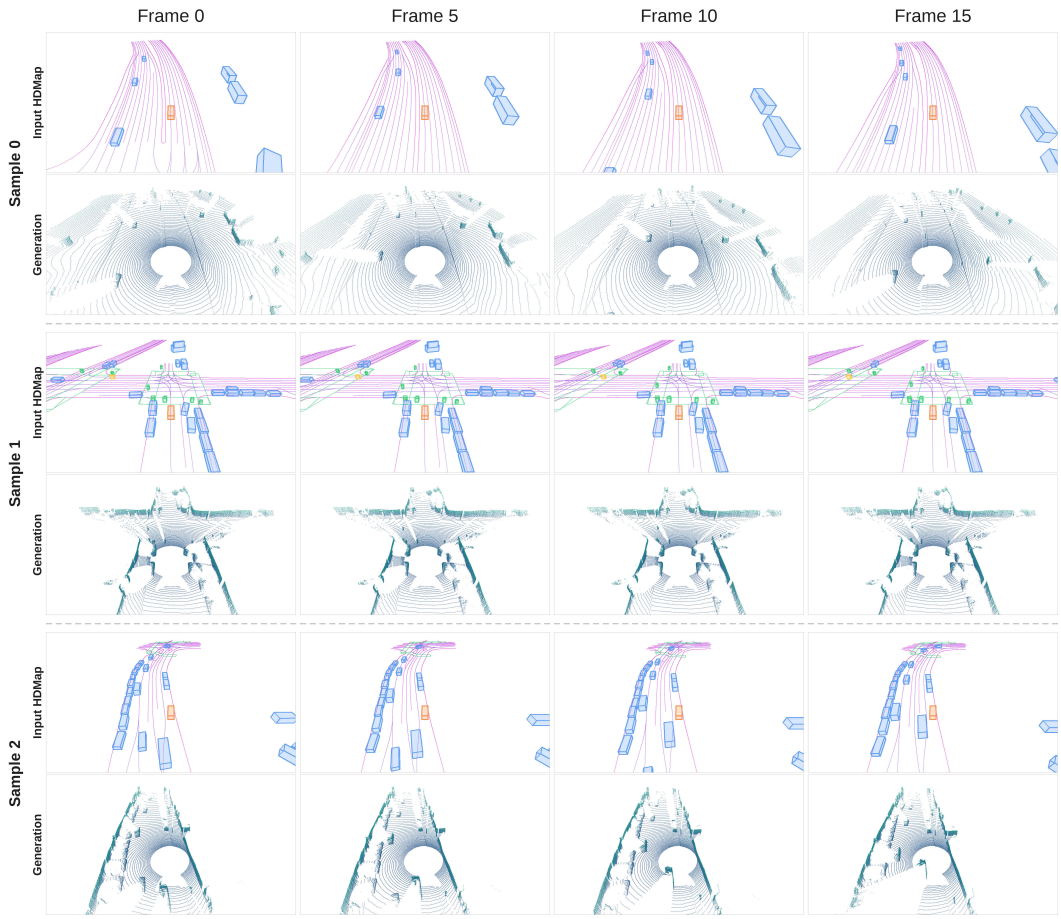

Figure 10: **HD-Map-conditioned generation with the bidirectional model.** Sample 0 depicts a high-speed highway scene. Sample 1 shows a busy intersection with pedestrians. Sample 2 contains a moving truck approaching from behind. All sequences are conditioned only on the HD-Map.

A.6 Additional Qualitative Results for the Causal Student

HD-Map-Conditioned Generation We present the qualitative results of the causal HD-Map-conditioned generation in Fig. 13. We observe that our generated sequences remain faithful to the input and stable over long-horizon rollouts, whereas baselines visibly degrade.

Generalizable Resimulation We present the qualitative results of generalizable resimulation in Fig. 15, where we show that LidarVerse can preserve captured scene structure in both interpolation and extrapolation views, whereas baselines show visible artifacts under the sparse input setting. For the protocol in Sec. 4.3, frames 0, 4, 9, and 14 of each 20-frame chunk form the observed input. Interpolation is evaluated on frames 0–14 excluding those inputs, extrapolation on frames 15–19, and aggregate results on all non-input frames. We report L_1 error, AbsRel, and CD, together with per-scene optimization time and rendering throughput. Sparse observed sweeps form the fixed frozen-dynamic PSC; no HD-Map or per-scene optimization is required for our model. An observed LiDAR prefix can optionally provide temporal context.

A.7 Additional Results on Generalization Across Sensors and Datasets

A general LiDAR representation should not be tied to the scanning pattern of one sensor or the distribution of one dataset. We therefore evaluate whether the attention-based tokenizer and shared generative interface can be adapted jointly across heterogeneous LiDAR configurations.

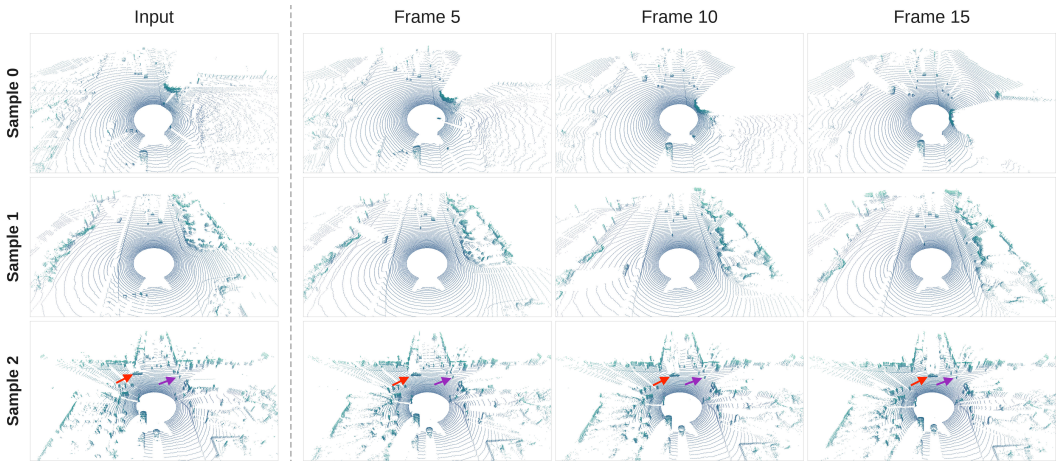

Figure 11: **First-frame-conditioned generation of the bidirectional model.** The red arrow indicates a moving car crossing the intersection, and the purple arrow indicates a pedestrian in the crosswalk. Sample 0 depicts a high-speed highway scene, Sample 1 urban traffic, and Sample 2 a crowded intersection with pedestrians crossing the road.

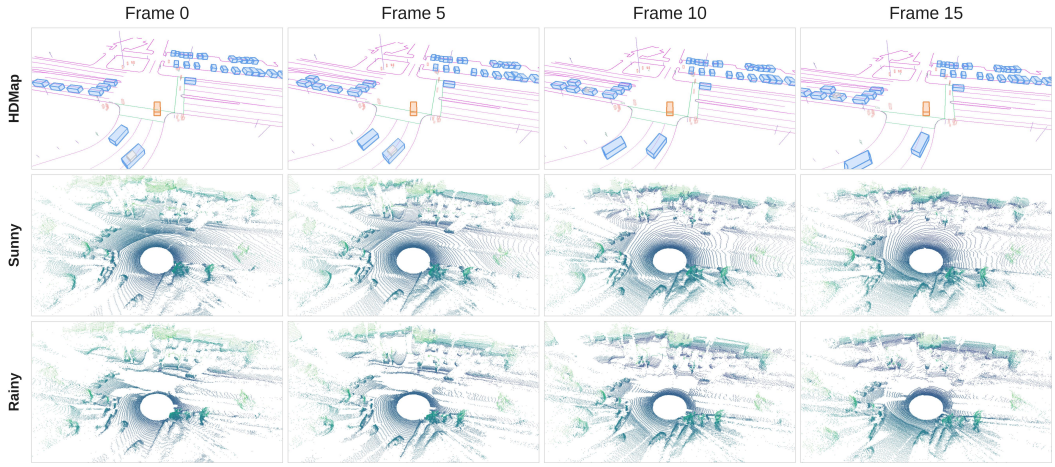

Figure 12: **Weather-controlled generation.** We show an example of our text-conditioned generation, where we rewrite the original sunny-weather prompt to request rain.

Cross-Dataset Reconstruction In Fig. 16, we show tokenizer reconstructions for all three training sensor configurations (Waymo, nuScenes, and NVIDIA PAI), following the protocol in Sec. 4.1, and demonstrate generalization to the unseen PandaSet (Xiao et al., 2021) scanning pattern. To further isolate the benefit of cross-dataset training, we compare its reconstruction quality on each dataset with that of a specialist tokenizer trained only on that dataset. For a fair comparison, all models use the same architecture and number of training iterations. Table 9 indicates that joint training across datasets improves reconstruction quality over each specialist.

Table 9: **Universal vs. specialist tokenizer reconstruction.** Metrics are in meters; lower is better.

Metric	Model	nuScenes	Waymo	NVIDIA PAI
$L_1 \downarrow$	Specialist	0.087	0.056	0.286
	Universal	**0.065**	**0.037**	**0.087**
CD $\downarrow$	Specialist	**0.557**	0.718	1.240
	Universal	0.568	**0.710**	**1.135**

Cross-Dataset Generation (with Intensity) To enable cross-sensor generation, we used the tokenizer above and jointly trained bidirectional model with additional sensor-specific embedding. In Fig. 17, we show that we can generate distinct nuScenes, NVIDIA PAI, and Waymo LiDAR patterns

from the same Waymo HD-Map input. We further extend cross-dataset reconstruction and generation to LiDAR intensity. We fine-tune the intensity-aware tokenizer on the same dataset mixture and then fine-tune the bidirectional teacher on the resulting range-intensity latents. The generated sequences in Fig. 18 exhibit plausible intensity profiles, including high-intensity returns from retroreflective surfaces.

A.8 Intensity Modeling

Sensor simulation must capture not only range geometry but also auxiliary measurements used by downstream perception. We therefore extend the model to LiDAR intensity, which is particularly important for tasks such as semantic segmentation. We fine-tune the tokenizer to reconstruct intensity and subsequently fine-tune the bidirectional teacher on the combined range-intensity latent representation. Using the resimulation protocol for top-20 travel distance scenes from Sec. 4.3, our intensity predictions remain competitive with those of the state-of-the-art reconstruction baselines (Tab. 10).

Table 10: **Generalizable LiDAR resimulation intensity RMSE.** Ours denotes the intensity-fine-tuned bidirectional teacher

Method	Interp.↓	Extrap.↓	Agg.↓
LiDAR4D	**0.077**	<u>0.094</u>	**0.082**
LiDAR-RT	0.083	0.100	0.089
SplatAD	0.115	0.134	0.121
Ours	<u>0.078</u>	**0.091**	**0.082**

A.9 Spatial-Cache Confirmation

With a fixed nine-frame K/V cache, we vary the number of distinct observations required to promote a candidate voxel over $N_{\min} \in \{1, 2, 3, 4, 5\}$. As shown in Fig. 19, directly registering the latest sweep after generation (i.e., $N_{\min} = 1$) produces worse fidelity metrics despite achieving the best temporal-consistency metrics. We choose $N_{\min} = 3$ to balance fidelity and temporal consistency.

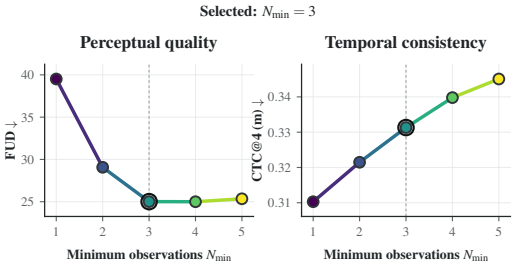

Figure 19: **Spatial-cache confirmation ablation.**

A.10 Limitations and Future Work

Currently, LidarVerse models a single spinning LiDAR. Future work could extend the framework to solid-state LiDAR and joint generation for multi-LiDAR rigs, leveraging datasets such as the Waymo Open Dataset (Sun et al., 2020) and Argoverse 2 (Wilson et al., 2021). Second, LidarVerse still relies on annotated object poses for dynamic resimulation during both training and inference, limiting its scalability to unlabeled scenes. While recent RGB-based methods have explored generalizable reconstruction without such object-level supervision (Yang et al., 2025; Wang et al., 2025; Lu et al., 2025), annotation-free generalizable resimulation remains an important open challenge for LiDAR. Finally, LidarVerse focuses exclusively on LiDAR. Extending the framework to jointly generate synchronized observations across sensor modalities, such as cameras and LiDAR, is a promising direction toward more comprehensive autonomous-driving simulation, following recent advances (Ren et al., 2025; Wang et al., 2026).

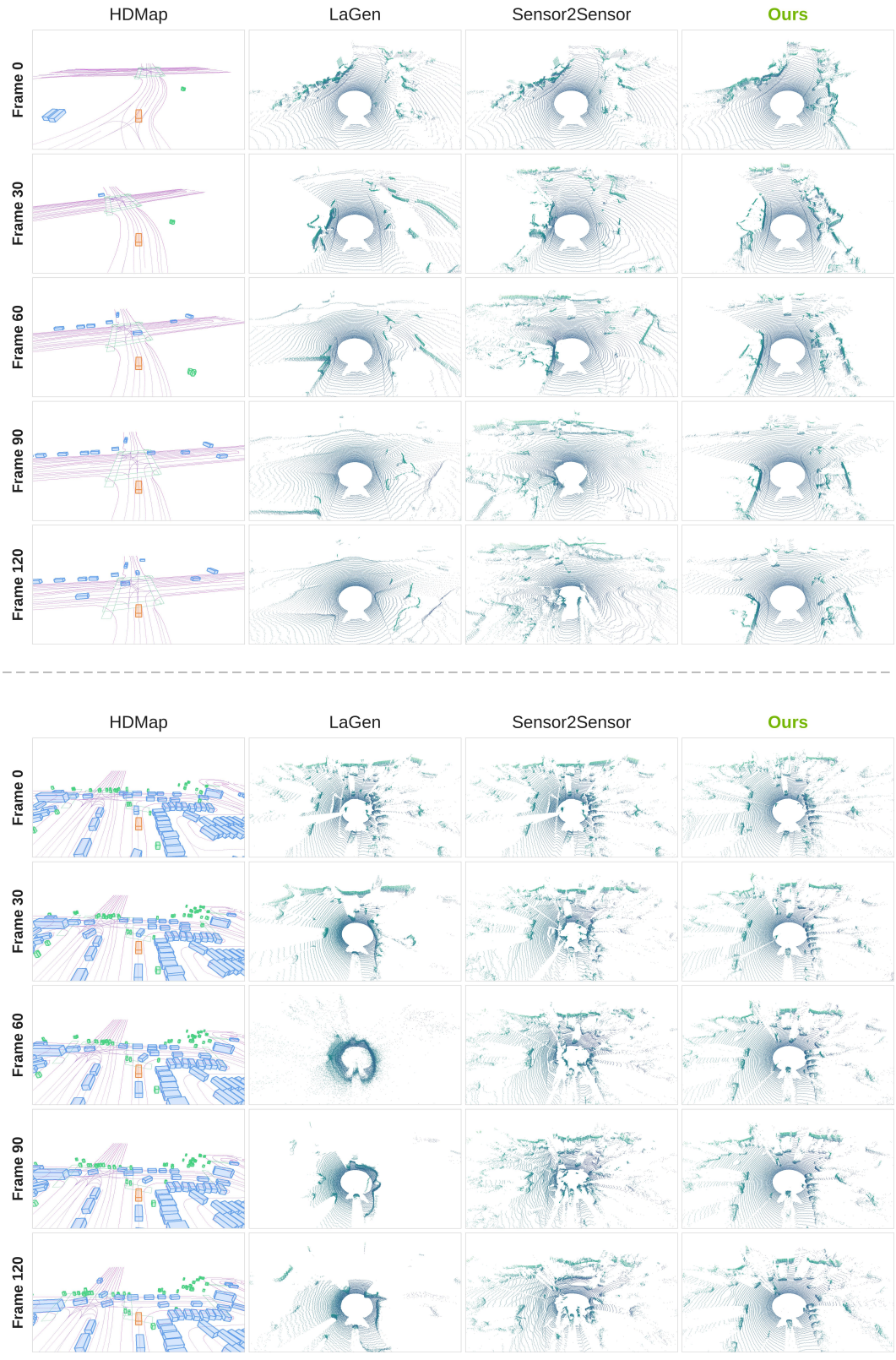

Figure 13: **Autoregressive LiDAR generation**: LidarVerse generates realistic and consistent LiDAR and is robust under long-horizon rollouts, whereas baselines collapse after 100 frames.

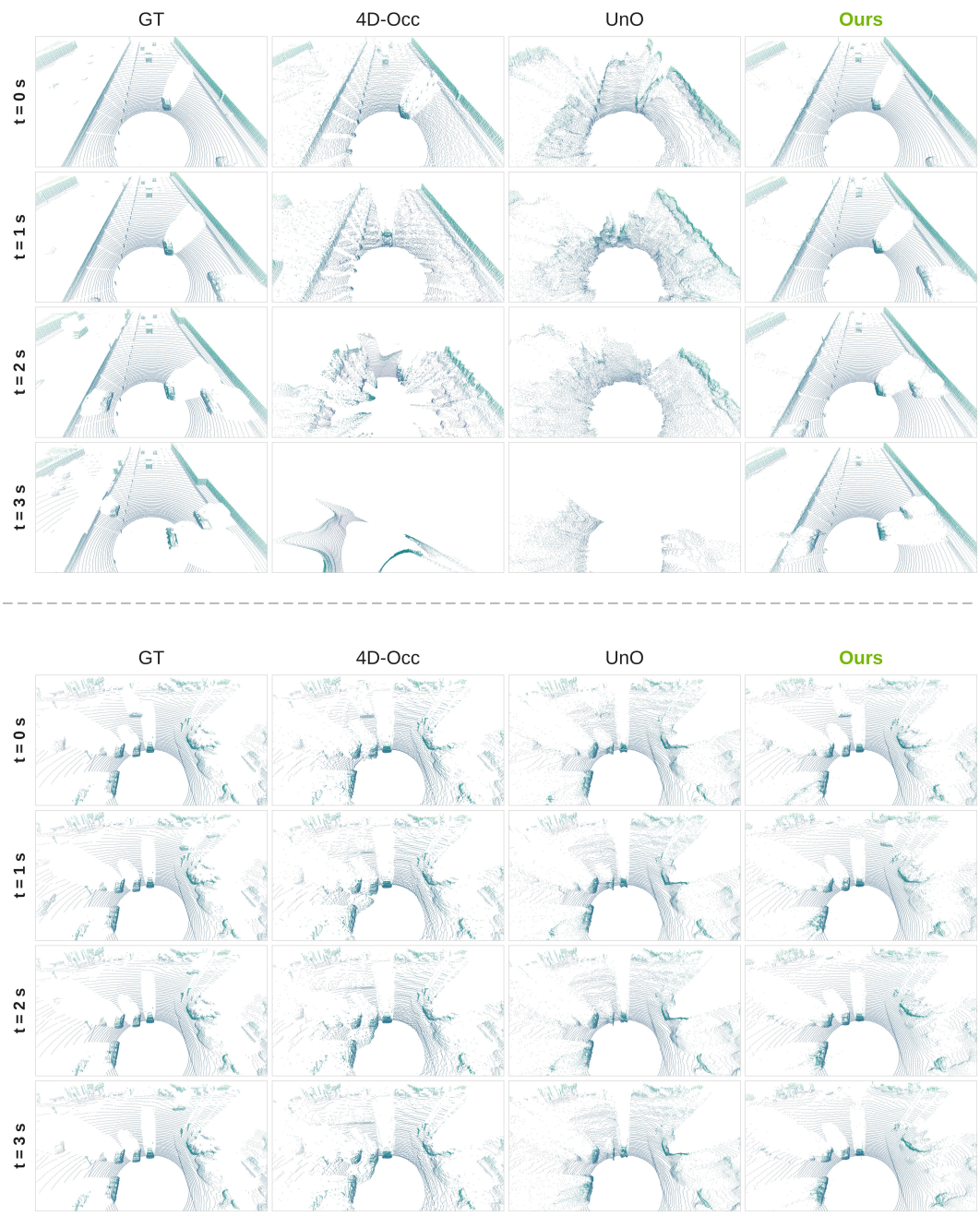

Figure 14: **Point-cloud forecasting**: LidarVerse preserves dynamic agents and synthesizes plausible geometry as the future viewpoint reveals regions not observed in the input.

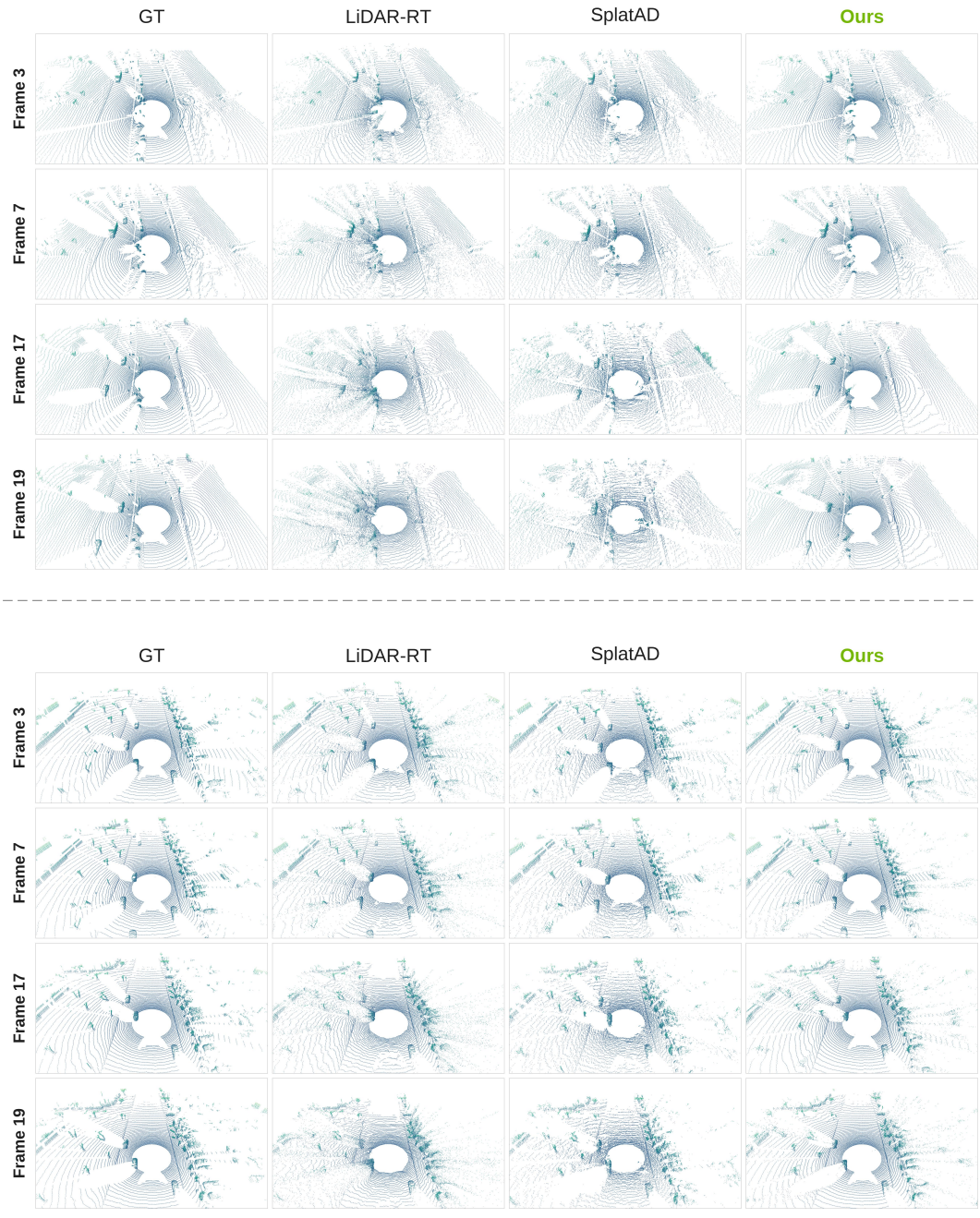

Figure 15: **Generalizable LiDAR resimulation**: the causal LidarVerse checkpoint achieves strong resimulation fidelity in both the interpolation setting (frames 3 and 7) and the extrapolation setting (frames 17 and 19). It requires no per-scene optimization and renders in real time.

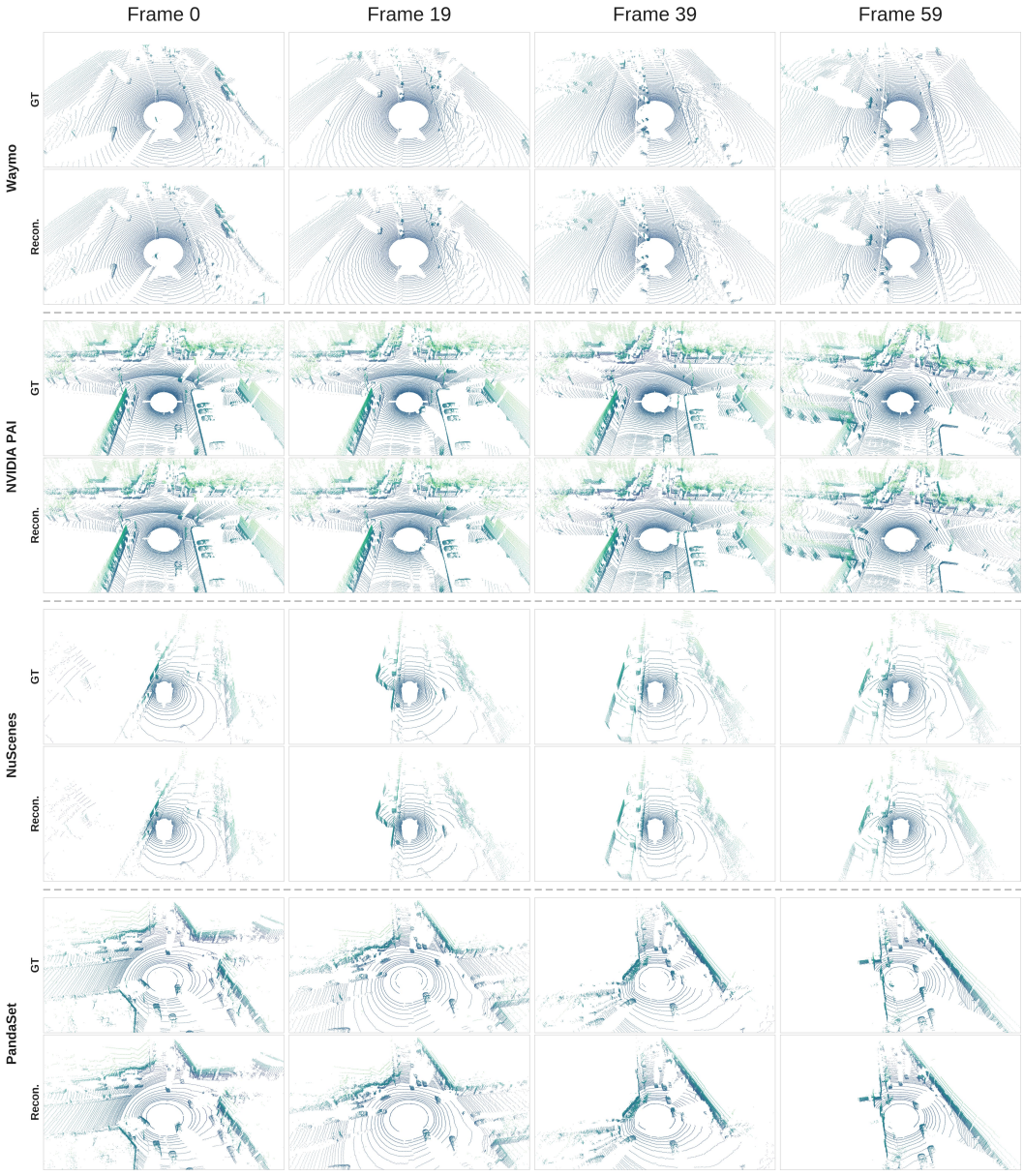

Figure 16: **Cross-dataset reconstruction.** One tokenizer handles 32-, 64-, and 128-beam LiDAR configurations and generalizes to PandaSet, which is not included in training.

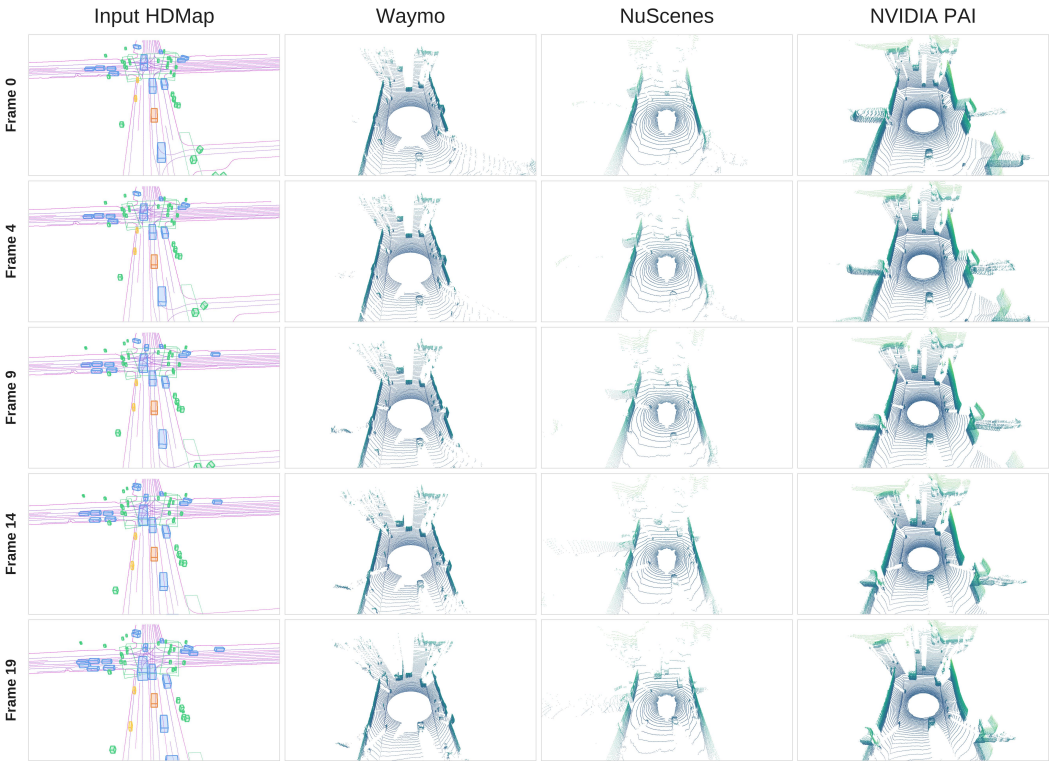

Figure 17: **Cross-dataset generation.** By varying the sensor embedding, our bidirectional teacher generates distinct LiDAR scanning patterns from the same Waymo HD-Map input.

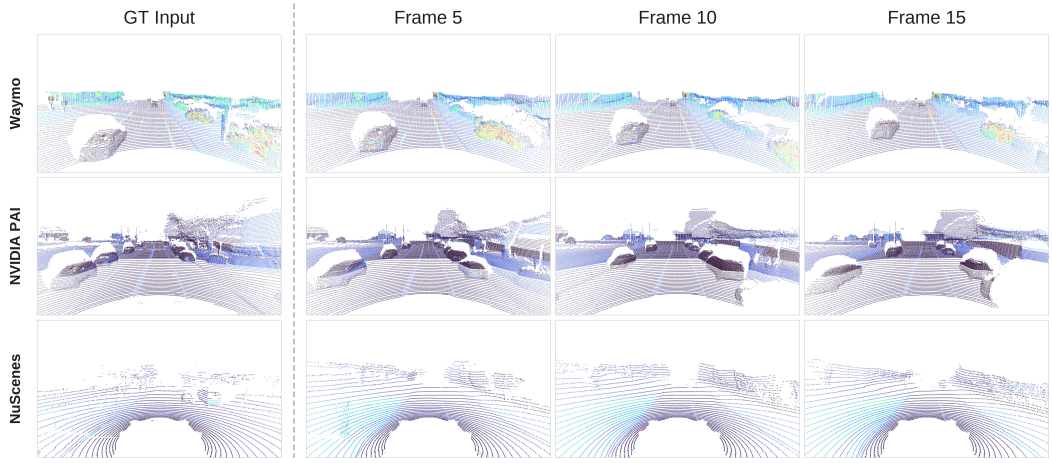

Figure 18: **Cross-dataset first-frame-conditioned generation with intensity.** Using the universal tokenizer, the bidirectional teacher generates high-quality LiDAR sequences with intensity across datasets within a single model. Each sequence is conditioned on its first ground-truth LiDAR frame.