LiDARCrafter: Dynamic 4D World Modeling from LiDAR Sequences

Alan Liang[1,2,3], **Youquan Liu**[4], **Yu Yang**[5], **Dongyue Lu**[1], **Linfeng Li**[1], **Lingdong Kong**[1,6,*], **Huaici Zhao**[3,†], **Wei Tsang Ooi**[1,†]

[1]National University of Singapore
[2]University of Chinese Academy of Sciences
[3]Shenyang Institute of Automation, Chinese Academy of Sciences
[4]Fudan University
[5]Zhejiang University
[6]CNRS@CREATE, Singapore

Abstract

Generative world models have become essential data engines for autonomous driving, yet most focus on videos or occupancy grids and overlook the unique challenges of LiDAR. Extending LiDAR generation to dynamic 4D modeling requires addressing controllability, temporal coherence, and standardized evaluation. We present **LiDARCrafter**, a unified framework for controllable 4D LiDAR generation and editing. Free-form language instructions are converted into ego-centric scene graphs that guide a tri-branch diffusion model to generate object geometry, motion, and structural priors. An autoregressive module further produces temporally coherent and stable LiDAR sequences with improved global consistency. To enable fair comparison, we introduce a comprehensive benchmark covering scene-, object-, and sequence-level metrics for rigorous and reproducible evaluation. Experiments on nuScenes show that **LiDARCrafter** achieves state-of-the-art fidelity, controllability, and temporal consistency, paving the way for scalable data augmentation and realistic simulation in diverse scenarios. Code have been publicly available at https://lidarcrafter.github.io.

1 Introduction

Generative world models are rapidly advancing the synthesis of large-scale sensor data for autonomous driving (Hu et al. 2023). Most recent efforts focus on structured modalities such as video or occupancy grids, whose dense and regular formats align well with image pipelines (Wang et al. 2024). In contrast, LiDAR, despite its importance for metric 3D geometry and all-weather reliability, remains underexplored. Its point clouds are sparse, unordered, and irregular (Kong et al. 2023b; Liang et al. 2025), making image- or grid-based generation techniques poorly transferable.

Early efforts, such as LiDARGen, project 360° scans to range images and borrow pixel-based methods (Zyrianov, Zhu, and Wang 2022). Later approaches improve single-frame fidelity but stop short of dynamics (Nakashima and Kurazume 2024). Multimodal systems like UniScene rely on occupancy as intermediaries, limiting LiDAR independence and increasing computation (Li et al. 2025). A dedicated 4D LiDAR world model is therefore still absent.

Addressing this challenge requires progress on three fronts. First, *controllability*: text prompts offer accessible interfaces but lack spatial specificity, whereas structured inputs (*e.g.*, boxes or trajectories) require costly annotation (Bian et al. 2025; Yang et al. 2025a). Second, *temporal consistency*: reliable downstream use requires modeling occlusions and object kinematics beyond single-frame generation. Third, *standardized evaluation*: unlike video models, LiDAR generation still lacks unified metrics for assessing fidelity and consistency across views (Huang et al. 2024).

To close these gaps, we introduce **LiDARCrafter**, a unified framework for controllable 4D LiDAR generation. At its core is an explicit, object-centric 4D layout that encodes geometry and motion while providing *precise yet accessible* control. **Text2Layout** parses natural-language instructions into an ego-centric scene graph and predicts object boxes, trajectories, and shape priors via a tri-branch diffusion model. **Layout2Scene** generates a high-fidelity initial scan from this layout using range-image diffusion, enabling fine-grained editing such as insertion, deletion, and dragging. **Scene2Seq** synthesizes the remaining frames *autoregressively*, warping past points with priors to ensure temporal coherence. We further introduce an *evaluation suite* that measures scene-, object-, and sequence-level quality.

Experiments on nuScenes (Caesar et al. 2020) show that **LiDARCrafter** achieves best single-frame fidelity, strong temporal consistency, and intuitive controllability, establishing a new benchmark for LiDAR-based 4D world modeling.

In summary, the core contributions of this work are:

- We present **LiDARCrafter**, the first 4D generative world model tailored to LiDAR, achieving superior controllability and spatiotemporal consistency.
- We propose a tri-branch, layout-conditioned pipeline for 4D layouts and precise LiDAR sequence generation.
- We introduce a comprehensive evaluation suite for 4D LiDAR and achieve leading performance on nuScenes.

2 Related Work

Driving Generative World Models. Generative world models aim to simulate scene dynamics for autonomous

*Project lead.
†Corresponding authors.

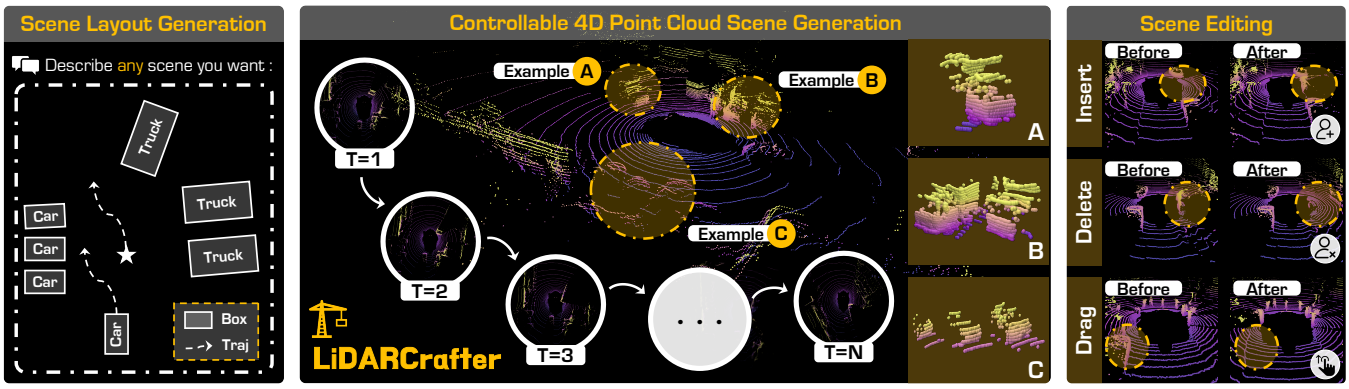

Figure 1: We propose **LiDARCrafter**, a 4D LiDAR-based generative world model that supports controllable point cloud layout generation (**left**), dynamic sequential scene generation (**center**), and rich scene editing applications (**right**). Our framework enables intuitive *"what you describe is what you get"* LiDAR-based 4D world modeling.

driving. Most recent efforts operate on video or occupancy representations. Video-based approaches such as GAIA-1 (Hu et al. 2023), DreamForge (Mei et al. 2024), and MagicDrive (Gao et al. 2023) leverage autoregressive modeling or BEV features for improved temporal consistency. Occupancy-centric methods, including OccWorld (Zheng et al. 2024a) and OccSora (Wang et al. 2024), provide structured spatial representations useful for downstream reasoning. Multimodal frameworks like UniScene (Li et al. 2025) and GENESIS (Guo et al. 2025) further align cross-modal cues for coherent generation. However, LiDAR-specific generative modeling remains underexplored, with prior efforts mostly focusing on forecasting or static scene synthesis (Zhang et al. 2023; Liu, Zhao, and Rhinehart 2025).

LiDAR Point Cloud Generation. Early LiDAR generative methods project point clouds into range images, as in LiDARGen (Zyrianov, Zhu, and Wang 2022). Recent diffusion-based approaches such as RangeLDM (Hu, Zhang, and Hu 2024), Text2LiDAR (Wu et al. 2024), and R2DM (Nakashima and Kurazume 2024) improve geometric fidelity through latent diffusion or single-stage denoising (Ho, Jain, and Abbeel 2020; Rombach et al. 2022). BEV-based pipelines like UltraLiDAR (Xiong et al. 2023) and OpenDWM (Ni et al. 2025) support richer scene editing, while cross-modal synthesis appears in X-Drive and UniScene (Li et al. 2025). Yet none of these works provide controllable *4D* LiDAR sequence generation with fine-grained temporal and object-level manipulation.

Controllability in Scene Synthesis. Controllable generation typically relies on structured inputs such as BEV semantic maps (Gao et al. 2023), HD maps (Swerdlow, Xu, and Zhou 2024), or 3D bounding boxes (Yang et al. 2024), though these require substantial annotation. Text-conditioned methods (Hu et al. 2023; Wu et al. 2024) offer more accessible interfaces but lack precise spatial grounding. Two-stage indoor synthesis frameworks (Zhai et al. 2024) and their outdoor extensions (Liu et al. 2025) demonstrate that intermediate scene graphs can enhance control. However, no existing approach supports dynamic, object-centric controllability for *4D LiDAR* scene generation.

3 LiDARCrafter: 4D LiDAR World Model

The cornerstone is an explicit 4D foreground layout that bridges the descriptive power of language and the geometric rigor required by LiDAR. As shown in Fig. 2, our framework adopts a three-stage process. In the **Text2Layout** stage (*cf.* Sec. 3.1), an LLM converts the instruction into an ego-centric scene graph, and a tri-branch diffusion sampler generates *object boxes, trajectories, and shape priors*, which serve as the conditioning layout signal. In the **Layout2Scene** stage (*cf.* Sec. 3.2), a range-image diffusion model turns the layout into a high-fidelity static scan. In the **Scene2Seq** stage (*cf.* Sec. 3.2), the static cloud is autoregressively warped and inpainted to yield drift-free frames. Finally, our **EvalSuite** (*cf.* Sec. 3.4) adds metrics for object semantics, layout soundness, and motion fidelity, giving the first comprehensive benchmark for 4D LiDAR generation.

3.1 Text2Layout: 4D Layout Generation

Natural-language prompts alone lack the spatial precision needed for complex world modeling. We therefore introduce a scene graph as an intermediate, explicit encoding of object geometry and relations. LLMs can parse text into such graphs, a strategy proven effective for scene synthesis (Yang et al. 2025b). **LiDARCrafter** extends this idea to dynamic outdoor settings. The LLM first builds a 4D scene graph from the prompt. A diffusion decoder then transforms this graph into a detailed layout of object boxes, trajectories, and shape priors that guide the LiDAR sequence generation.

Language-Driven Graph Construction. Given a textual instruction, we prompt an LLM (Achiam et al. 2023) to build an ego-centric scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A tailored query enumerates all foreground objects, producing the node set $\mathcal{V} = \{v_0, \ldots, v_M\}$, where v_0 denotes the ego vehicle and the remaining M nodes represent dynamic objects. Each node v_i is annotated with its semantic class c_i and a motion state phrase s_i (e.g., "go straight"). For every ordered pair (i, j) with $i \neq j$, a directed edge $e_{i \to j} \in \mathcal{E}$ encodes

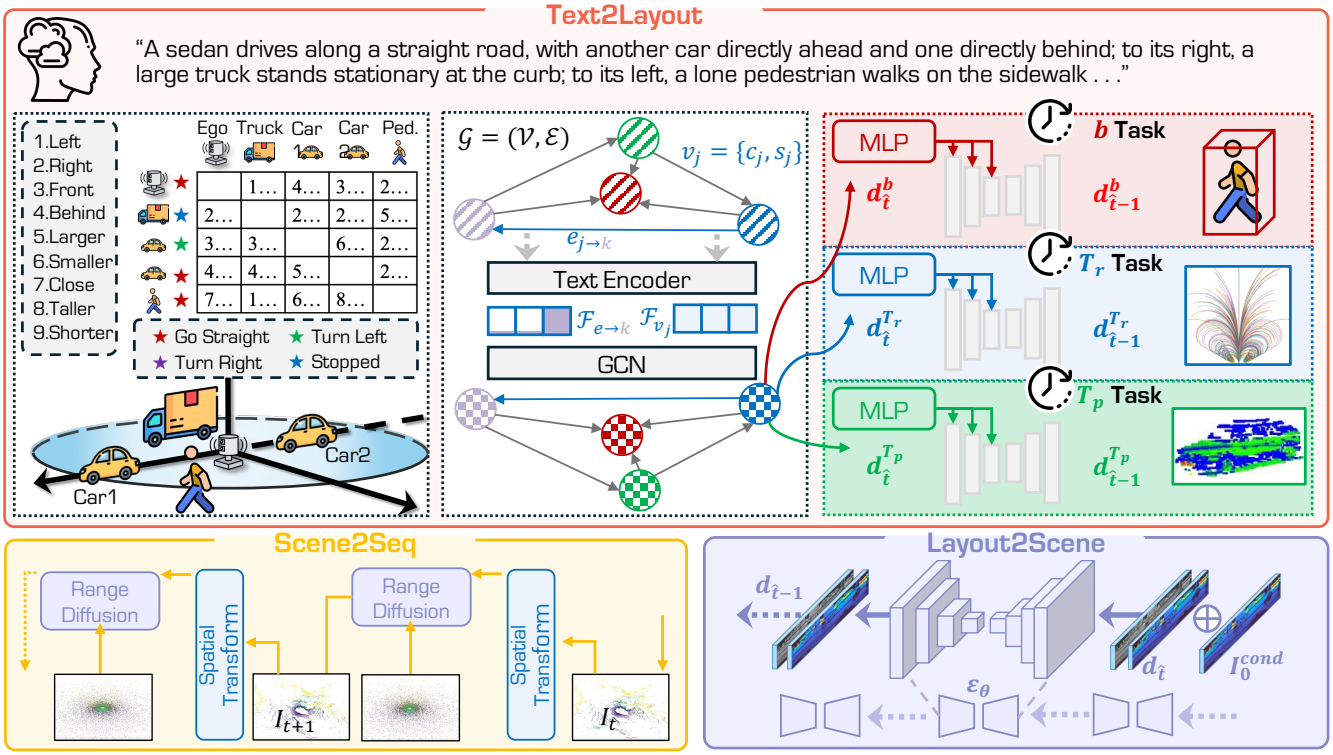

Figure 2: Framework of **LiDARCrafter**. In the **Text2Layout** stage (*cf.* Sec. 3.1), the natural-language instruction is parsed into an ego-centric scene graph, and a tri-branch diffusion network generates 4D conditions for bounding boxes, future trajectories, and object point clouds. In the **Layout2Scene** stage (*cf.* Sec. 3.2), a range-image diffusion model uses these conditions to generate a static LiDAR frame. In the **Scene2Seq** stage (*cf.* Sec. 3.3), an autoregressive module warps historical points with ego and object motion priors to generate each subsequent frame, producing a temporally coherent LiDAR sequence.

their spatial relation (e.g., "in front of", "lager than", details in Fig. 2). Unlike prior work (Liu et al. 2025), including the ego node yields a structurally complete scene graph that fully conditions downstream layout generation.

Scene-Graph Lifting. Given a textual scene graph, we aim to infer for each node v_i a 4D layout tuple $\mathcal{O}_i = (\mathbf{b}_i, \boldsymbol{\delta}_i, \mathbf{p}_i)$, where $\mathbf{b}_i = (x_i, y_i, z_i, w_i, l_i, h_i, \psi_i)$ is the 3D bounding box capturing the 3D center, size, and yaw angle of object i. $\boldsymbol{\delta}_i = \{(\Delta x_i^t, \Delta y_i^t)\}_{t=1}^{T}$ records planar displacements over T future frames, and $\mathbf{p}_i \in \mathbb{R}^{N \times 3}$ stores N canonical foreground points that sketch the shape of object. This tuple captures where, how, and what for every node and serves as the target of our denoiser during the diffusion process.

Graph-Fusion Encoder. To obtain context-aware priors for every tuple, following the method in the indoor area (Zhai et al. 2024), we process the scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an L-layer TripletGCN (Johnson, Gupta, and Fei-Fei 2018). We first embed nodes and edges with a frozen CLIP text encoder (Radford et al. 2021) to bring richer semantics:

$$\begin{aligned} \mathbf{h}_{v_i}^{(0)} &= \texttt{concat}(\text{CLIP}(c_i),\, \text{CLIP}(s_i),\, \boldsymbol{\omega}_i) \\ \mathbf{h}_{e_{i\to j}}^{(0)} &= \text{CLIP}(e_{i\to j}), \end{aligned} \tag{1}$$

where $\boldsymbol{\omega}_i$ is a learnable positional code. At layer ℓ, we update triplets with two lightweight MLPs: Φ_{edge} for edge reasoning and Φ_{agg} for neighborhood aggregation as follows:

$$\begin{aligned} (\tilde{\mathbf{h}}_{v_i}^{(\ell)}, \mathbf{h}_{e_{i\to j}}^{(\ell+1)}, \tilde{\mathbf{h}}_{v_j}^{(\ell)}) &= \Phi_{\text{edge}}(\mathbf{h}_{v_i}^{(\ell)}, \mathbf{h}_{e_{i\to j}}^{(\ell)}, \mathbf{h}_{v_j}^{(\ell)}) \\ \mathbf{h}_{v_i}^{(\ell+1)} &= \tilde{\mathbf{h}}_{v_i}^{(\ell)} + \Phi_{\text{agg}}\Big(\texttt{avg}\big(\tilde{\mathbf{h}}_{v_j}^{(\ell)} \mid v_j \in N_{\mathcal{G}}(v_i)\big)\Big). \end{aligned} \tag{2}$$

After L hops, each node feature $\mathbf{h}_{v_i}^{(L)}$ encodes both global semantics and local geometry, providing a strong semantic-geometric prior for LiDAR layout generation.

Layout Diffusion Decoder. The final node embeddings condition a tri-branch diffusion decoder (Rombach et al. 2022), one branch per element of $\mathcal{O}_i$. Let $\mathbf{d}_\tau^o$ be the noisy sample of modality $o \in \mathcal{O}_i$ at timestep τ. Each branch minimizes

$$\mathcal{L}^o = \mathbb{E}_{\tau, \mathbf{d}^o, \varepsilon} \big\| \varepsilon - \varepsilon_\theta^o(\mathbf{d}_\tau^o, \tau, c^o) \big\|_2^2, \tag{3}$$

sharing a common noise schedule.

Boxes and trajectories are denoised using a lightweight 1D U-Net (Ronneberger, Fischer, and Brox 2015), while object shapes are synthesised with a point-based U-Net (Zheng et al. 2024b). Unlike LOGen (Yan et al. 2025), we match only the LiDAR distribution, not the exact foreground points, which eliminates the heavy DiT cost (Peebles and Xie 2023) yet still delivers plausible inputs for refinement.

3.2 Layout2Scene: Controlled LiDAR Generation

LiDARCrafter ensures generation fidelity by using a unified range-image diffusion backbone that generates LiDAR

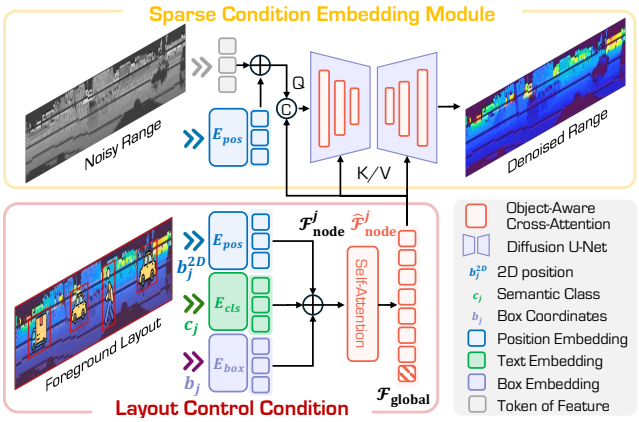

Figure 3: Details of our range-image diffusion model.

point clouds end to end. Given the scene graph $\mathcal{G}$ and the decoded layout $\mathcal{O}_i$, the network denoises Gaussian noise into the clean range frame $\mathbf{I}^0$, thereby bootstrapping the LiDAR sequence $\mathcal{P} = \{\mathbf{P}^t\}_{t=0}^{T}$ while following the pose, trajectory, and coarse shape of each object. The range-view representation preserves native LiDAR geometry while remaining convolution-friendly (Nakashima and Kurazume 2024; Kong et al. 2023a; Xu et al. 2025; Kong et al. 2025).

Sparse Object Conditioning. Directly projecting all foreground points into the range image, as in OLiDM (Yan et al. 2025), inadequately represents small or distant objects (e.g., a car at 15m may occupy only a few dozen pixels). To address this, we condition the model on sparse object representations that encode semantics, pose, and coarse shape, thereby enabling the model to hallucinate fine structure, as shown in Fig. 3. For each node, we aggregate its features

$$\hat{\mathbf{h}}_{v_i} = \Phi_{\text{pos}}\big(\pi(\mathbf{b}_i)\big) + \Phi_{\text{cls}}(c_i) + \Phi_{\text{box}}(\mathbf{b}_i), \tag{4}$$

where $\pi(\mathbf{b}_i)$ is the 3D box projected to image coordinates, Φ_{pos} is a positional embedder, and $\Phi_{\text{cls}}, \Phi_{\text{box}}$ are learned MLPs. A lightweight self-attention layer diffuses contextual cues across tokens (Vaswani et al. 2017), producing the refined vector $\mathbf{h}_{v_i}$. The ego token is further compressed by an MLP to form a scene-level vector $\mathbf{h}_{\text{ego}}$.

During denoising step τ, the noisy range map $\mathbf{d}_\tau$ is concatenated with a sparse conditioning map $\mathbf{I}_{\text{cond}}$ as model input, which is obtained by projecting all layout points $\{\mathbf{p}_i\}_{i=0}^{M}$ onto the image plane. The global context is formed by summing the scene-level vector, a time embedding, and a CLIP embedding of the ego state:

$$\mathbf{h}_{\text{cond}} = \mathbf{h}_{\text{ego}} + \Phi_{\text{time}}(\tau) + \text{CLIP}(s_0). \tag{5}$$

A transformer-based U-Net then predicts the clean signal, progressively sharpening geometry and semantics.

Layout-Driven Scene Editing. As each frame is anchored by an explicit layout, we can edit objects without disturbing the static background, which is crucial for testing planners. After the original scene $\mathbf{d}_0^{\text{orig}}$ is synthesized, a user may alter the layout tuple. We then rerun the reverse diffusion, preserving pixels whose 2D projections remain unchanged fol-

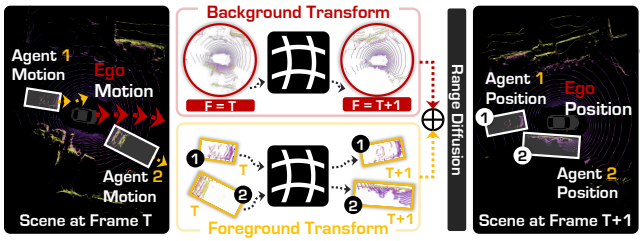

Figure 4: Details of the foreground and background warp.

lowing (Lugmayr et al. 2022). At each denoising step:

$$\mathbf{d}_{\tau-1} = (1-\mathbf{m}) \odot \tilde{\mathbf{d}}_{\tau-1} + \mathbf{m} \odot \hat{\mathbf{d}}_{\tau-1}, \tag{6}$$

where $\hat{\mathbf{d}}_{\tau-1}$ is the freshly denoised sample, $\tilde{\mathbf{d}}_{\tau-1} \sim \mathcal{N}\big(\sqrt{\bar{\alpha}}\,\mathbf{d}_0^{\text{orig}}, (1-\bar{\alpha})\mathbb{I}\big)$ is a Gaussian-perturbed copy of the original scene, and the binary mask $\mathbf{m}$ marks pixels affected by the edited boxes. The blend locks untouched regions and resynthesizes only the modified objects, delivering instant, artifact-free edits for closed-loop simulation.

3.3 Scene2Seq: Autoregressive LiDAR Synthesis

A core innovation of **LiDARCrafter** is its ability to generate the LiDAR stream autoregressively. In RGB video, textures and lighting change constantly, whereas a LiDAR sweep sees a mostly static environment, with only the ego vehicle and annotated objects moving. We exploit this stability by warping previously observed points to create a strong prior, as shown in fig. 4. Concretely, we back-project the first range image $\mathbf{I}^0$ to a point cloud $\mathbf{P}^0$, then split it with the layout boxes into background $\mathbf{B}^0$ and foreground sets $\{\mathbf{F}_i^0\}_{i=1}^{M}$. In later frames, we warp $\mathbf{B}^0$ with the ego pose, and update each $\mathbf{F}_i^0$ by its own motion prior, providing a strong drift-free geometric prior for the diffusion model at every denoising step.

Static-Scene Warp. We update the background points with the ego pose. Taking frame 0 as the world origin, the ego translation at step t is $\mathbf{u}_0^t = [\Delta x_0^t, \Delta y_0^t, z_0]^\top$, with z_0 is the fixed sensor height, and its incremental yaw is

$$\psi_0^t = \text{atan2}(\Delta y_0^t - \Delta y_0^{t-1},\ \Delta x_0^t - \Delta x_0^{t-1}). \tag{7}$$

We form the homogeneous ego pose matrix $\mathbf{G}_0^t \in \text{SE}(3)$ with the rotation matrix $\mathbf{R}_z(\psi_0^t)$ and translation $\mathbf{u}_0^t$, and compute the relative motion $\Delta\mathbf{G}_0^t = \mathbf{G}_0^t(\mathbf{G}_0^{t-1})^{-1}$, then we propagate the static cloud via $\mathbf{B}^t = \Delta\mathbf{G}_0^t\mathbf{B}^{t-1}$.

Dynamic-Object Warp. For each object i, we first shift its box center by its own cumulative planar offsets $(\Delta x_i^t, \Delta y_i^t)$, giving the world-frame position $\mathbf{u}_i^t = [x_i + \Delta x_i^t,\ y_i + \Delta y_i^t,\ z_i]^\top$. Its heading change ψ_i^t is obtained exactly as in eq. (7). To express the box in the current ego frame, we apply the inverse ego transform: first translate by $-\mathbf{u}_0^t$ and then rotate by $-\psi_0^t$. The same rigid transform maps the stored foreground points $\mathbf{F}_i^0$ to $\mathbf{F}_i^t$. These warped foreground object points, combined with the updated background points, supply a strong geometric prior for the later timestep.

Autoregressive Generation. At every timestep $t > 0$ we build a condition range map by projecting and combining,

$$I_{\text{cond}}^t = \Pi\big(\mathbf{B}^{0\to t} \cup \mathbf{B}^{t-1\to t} \cup \{\mathbf{F}_i^{t-1\to t}\}_{i=1}^{M}\big), \tag{8}$$

Method	Range		Points		BEV	
	FRD↓	MMD↓	FPD↓	MMD↓	JSD↓	MMD↓
UniScene	–	–	976.47	29.06	31.55	13.61
OpenDWM	–	–	714.19	21.95	20.17	5.61
OpenDWM-DiT	–	–	381.91	12.46	19.90	5.73
LiDARGen	759.65	1.71	159.35	35.52	5.74	2.39
LiDM	495.54	0.18	210.20	8.45	5.86	0.73
RangeLDM	–	–	–	–	5.47	1.92
R2DM	243.35	1.40	33.97	1.62	3.51	0.71
LiDARCrafter	**194.37**	**0.08**	**8.64**	**0.90**	**3.11**	**0.42**

Table 1: Evaluations of scene-level fidelity for LiDAR generation on the *nuScenes* dataset. MMD values are reported in 10^{-4} and JSD in 10^{-2}. Lower is better for all metrics (↓).

#	Method	Venue	Car↑	Ped↑	Truck↑	Bus↑	#Box
Uncond.	LiDARGen	ECCV'22	0.57	0.29	0.42	0.38	0.364
	LiDM	CVPR'24	0.65	0.22	0.45	0.31	0.28
	R2DM	ICRA'24	0.54	0.29	0.39	0.35	0.53
Cond.	UniScene	CVPR'25	0.53	0.28	0.35	0.25	0.98
	OpenDWM	CVPR'25	0.74	0.30	0.51	0.44	0.54
	OpenDWM-DiT	CVPR'25	0.78	0.32	**0.56**	0.51	0.64
	LiDARCrafter	**Ours**	**0.83**	**0.34**	0.55	**0.54**	**1.84**

Table 2: Comparison of foreground object quality using FDC (↑), which reflects detector confidence on generated scenes. #Box is the average number of boxes per frame.

where $\Pi(\cdot)$ denotes spherical projection, and the superscript indicates the warp between two timestamps. Including the first frame background warp $\mathbf{B}^{0\to t}$ eliminates accumulated drift. We concatenate I^t_{cond} with the noisy sample and feed it into the diffusion backbone to generate the next range image, iterating until the whole sequence is synthesized.

3.4 EvalSuite: Temporal & Semantic Scoring

Existing LiDAR generation metrics like FRD judge only static realism. They ignore object semantics, layout validity, and motion coherence, which are essential for a controllable 4D world model. Our EvalSuite adds targeted scores for each facet. **Object metrics** (FDC, CDA, CFCA, CFSC) verify that generated foreground clouds carry the right labels, box geometry, and detector confidence. **Layout metrics** (SCR, MSCR, BCR, TCR) measure spatial and trajectory consistency while penalizing box or path collisions. **Temporal metrics** (TTCE, CTC) track frame-to-frame transform accuracy and sequence smoothness. Together, these metrics give a complete, 4D-aware assessment. More details are given in the supplementary material.

4 Experiments

4.1 Experimental Settings

We evaluate **LiDARCrafter** on the nuScenes dataset (Caesar et al. 2020). Evaluation combines standard static metrics (FRD, FPD, JSD, MMD) with our object-, layout-, and motion-centric scores (Sec. 3.4). Implementation and training details are provided in the supplementary material.

Method	Venue	$\text{AP}^{\text{R11}}_{\text{BEV}}$	$\text{AP}^{\text{R11}}_{\text{3D}}$	$\text{AP}^{\text{R40}}_{\text{BEV}}$	$\text{AP}^{\text{R40}}_{\text{3D}}$
UniScene	CVPR'25	0.19	0	0.02	0
OpenDWM	CVPR'25	17.07	9.09	11.84	1.03
OpenDWM-DiT	CVPR'25	16.37	11.27	10.62	1.89
LiDARCrafter	**Ours**	**23.21**	**15.24**	**18.27**	**8.26**

Table 3: Comparisons of foreground object perception accuracy using the CDA (↑) metric, which measures 3D detection average precision (AP) on generated LiDAR scenes.

#	Method	Venue	FPD↓	P-MMD↓	JSD↓	MMD↓
Uncond.	LiDARGen	ECCV'22	1.39	0.15	0.20	16.22
	LiDM	CVPR'24	1.41	0.15	0.19	13.49
	R2DM	ICRA'24	1.40	0.15	0.17	12.76
Cond.	UniScene	CVPR'25	1.19	0.18	0.23	16.65
	OpenDWM	CVPR'25	1.49	0.19	0.16	9.11
	OpenDWM-DiT	CVPR'25	1.48	0.18	**0.15**	9.02
	LiDARCrafter	**Ours**	**1.03**	**0.13**	**0.15**	**5.48**

Table 4: Evaluation of object-level fidelity for LiDAR generation. MMD is reported in 10^{-4}, and JSD in 10^{-2}.

4.2 Scene-Level LiDAR Generation

We evaluate scene-level LiDAR generation quality in terms of whole-scene fidelity and foreground object accuracy.

Whole-Scene Fidelity. As shown in Table 1, **LiDARCrafter** consistently outperforms prior methods on all scene-level metrics, achieving the lowest FRD and FPD. Qualitative results in Fig. 5 further show that our method produces scans closest to the ground truth, with cleaner backgrounds and better-preserved foreground structures.

Foreground Object Accuracy. We assess foreground quality using a pre-trained VoxelRCNN detector (Deng et al. 2021). **LiDARCrafter** achieves the highest Foreground Detection Confidence (FDC) and Conditioned Detection Accuracy (CDA) across most categories (Tables 2 and 3), indicating stronger alignment between generated objects and conditioning layouts.

4.3 Object-Level LiDAR Generation

This section evaluates the quality of individual object generation, focusing on both *fidelity* and *semantic and geometric consistency* under box-level conditioning.

Object-Wise Fidelity. To assess instance-level fidelity, we extract 2,000 *Car* objects from each method and compute object-level metrics (Table 4). LiDARCrafter achieves the lowest FPD (1.03) and MMD (5.48), significantly outperforming OpenDWM and demonstrating better reconstruction of fine-grained geometry.

Semantic and Geometric Consistency. To further evaluate object quality under conditioning, we introduce two metrics: CFCA for semantic fidelity and CFSC for geometric consistency (Table 5). For *semantic fidelity*, we apply a PointMLP (Ma et al. 2022) classifier trained on real data to classify generated instances, yielding a CFCA score of 73.48% for LiDARCrafter, indicating strong alignment with real-world categories. For *geometric consistency*, we use a conditional variational autoencoder to regress bounding

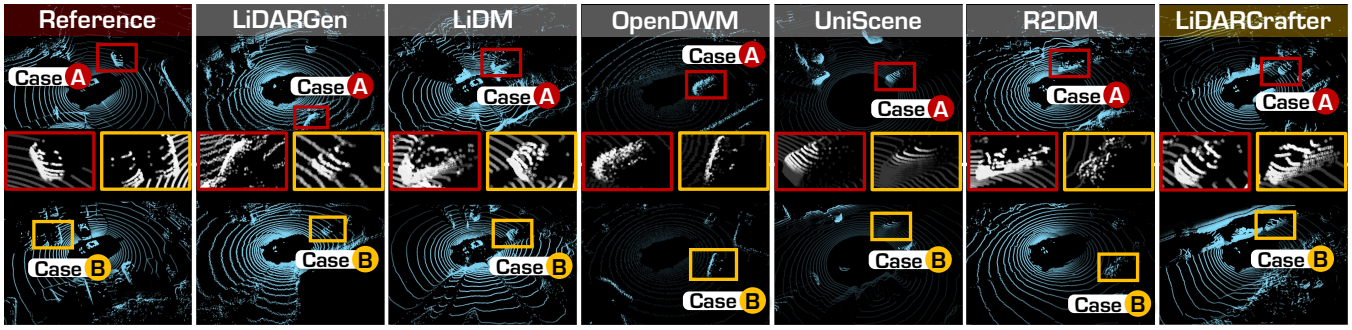

Figure 5: Single-frame LiDAR point cloud generation results. LiDARCrafter produces the pattern closest to the ground truth, with notably superior foreground quality compared to other methods. Best viewed at high resolution.

Method	Venue	CFCA↑	CFSC↑		
			< 150	150–300	> 300
Original	–	92.49	0.50	0.61	0.72
UniScene	CVPR'25	34.25	0.14	0.17	0.23
OpenDWM	CVPR'25	62.35	0.17	0.21	0.26
OpenDWM-DiT	CVPR'25	70.65	0.31	0.32	0.34
LiDARCrafter	**Ours**	**73.45**	**0.35**	**0.36**	**0.42**

Table 5: Comparison of object generation consistency using CFCA (↑) and CFSC (↑). CFCA measures classification accuracy on generated points using a PointMLP trained on real data. CFSC assesses geometric consistency by regressing boxes from generated points and computing IoU; the number indicates the point count within each box.

Method	Venue	TTCE↓		CTC↓			
		3	4	1	2	3	4
UniScene	CVPR'25	2.74	3.69	0.90	1.84	3.64	**3.90**
OpenDWM	CVPR'25	2.68	3.65	1.02	2.02	3.37	5.05
OpenDWM-DiT	CVPR'25	2.71	3.66	**0.89**	**1.79**	3.06	4.64
LiDARCrafter	**Ours**	**2.65**	**3.56**	1.12	2.38	**3.02**	4.81

Table 6: Comparison of temporal consistency in 4D LiDAR generation. Numbers indicate frame intervals.

boxes from generated point clouds, and compute the mean IoU with ground truth. LiDARCrafter achieves the highest IoU across all point count settings, demonstrating superior adherence to geometric constraints.

4.4 Autoregressive 4D LiDAR Generation

Temporal Consistency. We evaluate temporal consistency in 4D LiDAR generation in Table 6. TTCE measures the error between the predicted and ground-truth transformation matrices obtained via point cloud registration, while CTC computes the Chamfer Distance between consecutive frames. Our approach achieves the lowest TTCE scores across both frame intervals and maintains competitive CTC performance at all intervals, demonstrating strong temporal coherence. Qualitative comparisons in Figure 6 further show that LiDARCrafter produces sequences with consistent structure and fine geometric detail, whereas other methods often suffer from degraded fidelity over time.

No.	Type	Variant	Scene		Object		
			FRD↓	FPD↓	FPD↓	CFCA↑	CFSC↑
1	Baseline	–	243.35	33.97	1.40	–	–
2	Dense	$w/$ 2D mask	237.17	33.21	1.35	61.22	0.24
3		$w/$ Obj mask	217.83	24.02	1.20	64.54	0.27
4	Dense+Sparse	$w/$ E_{pos}	205.27	15.97	1.08	72.46	0.40
5		$w/$ $E_{pos} + E_{cls}$	**193.27**	10.52	1.05	**75.27**	0.40
6		$w/$ All	194.37	**8.64**	**1.03**	73.45	**0.42**

Table 7: Ablation on foreground conditioning methods for LiDAR generation. 2D masks are projected from 3D boxes.

No.	Type	Intensity	Depth	TTCE↓		CTC↓		Scene	
				3	4	3	4	FRD↓	FPD↓
1	E2E	-	-	3.21	4.36	5.68	7.41	477.21	182.36
2	AR	-	-	3.31	4.84	4.31	6.21	311.27	90.10
3		✓	✓	2.96	3.87	3.24	4.85	254.39	22.20
4		✓	✗	3.21	4.21	3.42	5.19	364.27	154.21
5		✗	✓	**2.65**	**3.56**	**3.02**	**4.81**	**194.37**	**8.64**

Table 8: Ablation on generation paradigm (E2E vs. AR) and historical conditioning for 4D LiDAR generation.

4.5 Ablation Study

We conduct ablations on **foreground generation** (necessity and conditioning) and **4D consistency** (generation paradigm and historical priors) to validate key designs.

Ablation on the necessity of foreground generation. Table 7 shows that introducing a 2D foreground mask projected from 3D boxes *(No.2)* notably improves scene generation, particularly for foreground objects. Further incorporating the foreground generation branch *(No.3)* that produces fine-grained object masks leads to a lower FPD, showing the benefit of detailed geometric and depth supervision.

Ablation on foreground conditioning mechanism. Foreground objects are inherently sparse, often occupying only a few pixels, making dense mask-only conditioning insufficient. As shown in Table 7, our proposed **sparse conditioning modules** are crucial: embedding 2D box features alone *(No.4)* reduces FRD, while adding semantic and geometric attributes *(No.5 and No.6)* yields the best FPD and further improves FRD. These results underscore the benefits of richer, object-centric conditioning.

Ablation on generation paradigm in 4D generation. Unlike RGB videos, where appearance varies due to light-

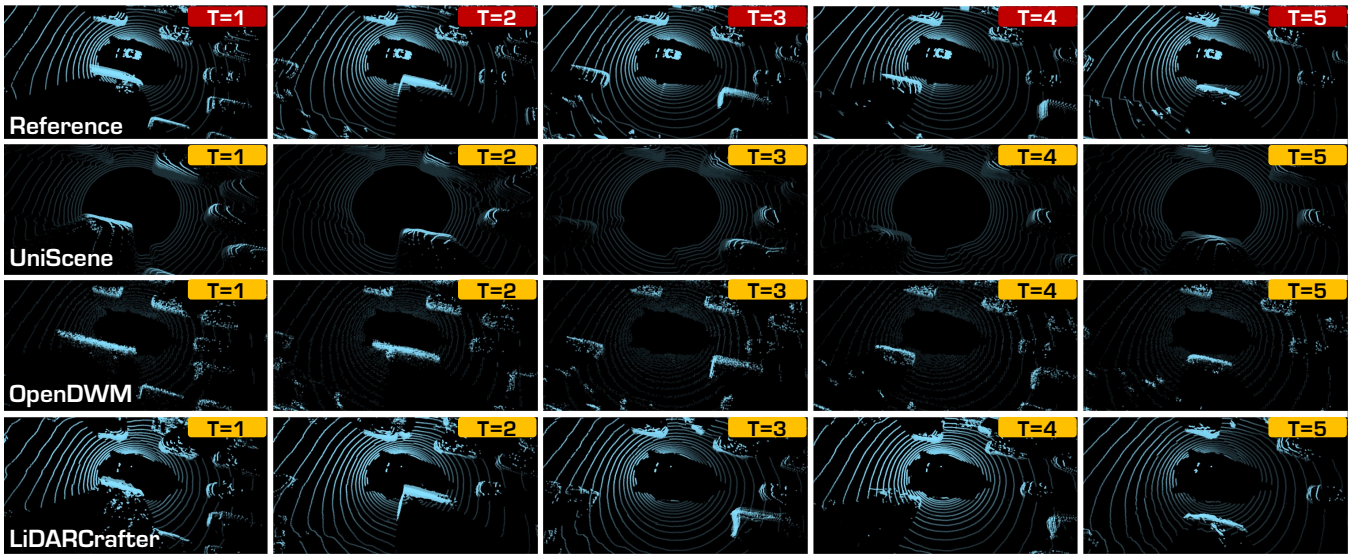

Figure 6: Sequence point cloud generation results. LiDARCrafter maintains temporal consistency while producing patterns closest to the ground truth. Frames are arranged in temporal order from left to right. Best viewed at high resolution.

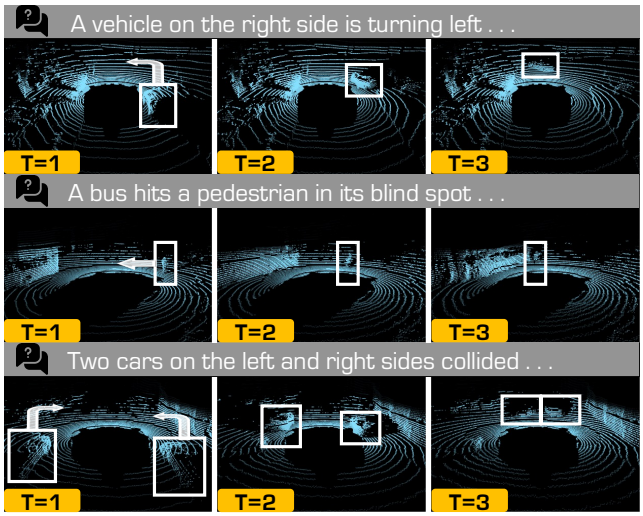

Figure 7: **Diverse corner cases** generated by LiDARCrafter with object-centric controllability. Best viewed at high resolution. Frames are arranged sequentially from left to right.

ing and texture changes, LiDAR sequences capture largely static environments, with dynamics introduced only by ego-motion and moving agents. We exploit this stability by warping previously observed points using ego and object trajectories, providing strong priors for **autoregressive generation**. As shown in Table 8, our inpainting-based autoregressive framework *(No.2)* outperforms the end-to-end baseline *(No.1)* on temporal metrics, demonstrating that the autoregressive design naturally aligns with the relatively static nature and limited temporal variation of LiDAR sequences.

Ablation on historical conditioning in 4D generation. Table 8 investigates the impact of different historical priors on 4D LiDAR generation. Using both depth and intensity features as conditioning inputs *(No.3)* significantly improves performance over the baseline without historical guidance *(No.2)*. Notably, excluding the depth prior *(No.4)* leads to substantial error accumulation (FRD increases by 109.88 compared to *No.3*), while using depth alone *(No.5)* achieves the best FRD. These results indicate that depth cues are more reliable and crucial for maintaining temporal consistency, whereas intensity features are harder to model effectively.

4.6 Applications

Leveraging its object-centric generation capability, **LiDARCrafter** enables the synthesis of rare and diverse corner-case scenarios for data augmentation and robustness evaluation. As shown in Fig. 7, our method generates challenging situations such as lane cut-ins, blind-spot pedestrians, vehicle collisions, and overtaking maneuvers, while maintaining strong temporal coherence. Additional qualitative results are provided in the supplementary material.

5 Conclusion

We presented **LiDARCrafter**, a unified framework for controllable 4D LiDAR sequence generation and editing. By leveraging scene graph descriptors, the multi-branch diffusion model, and an autoregressive generation strategy, our approach achieves fine-grained controllability and strong temporal consistency. Experiments on nuScenes demonstrate clear improvements over existing methods in fidelity, coherence, and controllability. Beyond high-quality data synthesis, LiDARCrafter enables the creation of safety-critical scenarios for robust evaluation of downstream autonomous driving systems. Future work will explore multi-modal extensions and further efficiency improvements.

Acknowledgments

This work is under the programme DesCartes and is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. This work is also supported by the Liaoning Applied Basic Research Program Fund under the project name: Research on 3D Target Detection and Tracking Methods for Intelligent Assisted Driving Applications (No. 2023JH2/101300239). Lingdong is supported by the Apple Scholars in AI/ML Ph.D. Fellowship program.

The author, Ao Liang, gratefully acknowledges the financial support from the China Scholarship Council.

Additionally, the authors would like to sincerely thank the Program Chairs, Area Chairs, and Reviewers for the time and effort devoted during the review process.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bian, H.; Kong, L.; Xie, H.; Pan, L.; Qiao, Y.; and Liu, Z. 2025. DynamicCity: Large-scale 4D occupancy generation from dynamic scenes. In *International Conference on Learning Representations*.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuScenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11621–11631.

Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2021. Voxel R-CNN: Towards high performance voxel-based 3D object detection. In *AAAI Conference on Artificial Intelligence*, 1201–1209.

Gao, R.; Chen, K.; Xie, E.; Hong, L.; Li, Z.; Yeung, D.-Y.; and Xu, Q. 2023. MagicDrive: Street view generation with diverse 3D geometry control. In *International Conference on Learning Representations*.

Guo, X.; Wu, Z.; Xiong, K.; Xu, Z.; Zhou, L.; Xu, G.; Xu, S.; Sun, H.; Wang, B.; Chen, G.; et al. 2025. Genesis: Multimodal driving scene generation with spatio-temporal and cross-modal consistency. *arXiv preprint arXiv:2506.07497*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Hu, A.; Russell, L.; Yeo, H.; Murez, Z.; Fedoseev, G.; Kendall, A.; Shotton, J.; and Corrado, G. 2023. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*.

Hu, Q.; Zhang, Z.; and Hu, W. 2024. RangeLDM: Fast realistic LiDAR point cloud generation. In *European Conference on Computer Vision*, 115–135. Springer.

Huang, Z.; He, Y.; Yu, J.; Zhang, F.; Si, C.; Jiang, Y.; Zhang, Y.; Wu, T.; Jin, Q.; Chanpaisit, N.; et al. 2024. VBench: Comprehensive benchmark suite for video generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21807–21818.

Johnson, J.; Gupta, A.; and Fei-Fei, L. 2018. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1219–1228.

Kong, L.; Liu, Y.; Chen, R.; Ma, Y.; Zhu, X.; Li, Y.; Hou, Y.; Qiao, Y.; and Liu, Z. 2023a. Rethinking range view representation for LiDAR segmentation. In *IEEE/CVF International Conference on Computer Vision*, 228–240.

Kong, L.; Liu, Y.; Li, X.; Chen, R.; Zhang, W.; Ren, J.; Pan, L.; Chen, K.; and Liu, Z. 2023b. Robo3D: Towards robust and reliable 3D perception against corruptions. In *IEEE/CVF International Conference on Computer Vision*, 19994–20006.

Kong, L.; Xu, X.; Ren, J.; Zhang, W.; Pan, L.; Chen, K.; Ooi, W. T.; and Liu, Z. 2025. Multi-modal data-efficient 3D scene understanding for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5): 3748–3765.

Li, B.; Guo, J.; Liu, H.; Zou, Y.; Ding, Y.; Chen, X.; Zhu, H.; Tan, F.; Zhang, C.; Wang, T.; et al. 2025. UniScene: Unified occupancy-centric driving scene generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11971–11981.

Liang, A.; Kong, L.; Lu, D.; Liu, Y.; Fang, J.; Zhao, H.; and Ooi, W. T. 2025. Perspective-invariant 3D object detection. In *IEEE/CVF International Conference on Computer Vision*.

Liu, T.; Zhao, S.; and Rhinehart, N. 2025. Towards foundational LiDAR world models with efficient latent flow matching. *arXiv preprint arXiv:2506.23434*.

Liu, Y.; Li, X.; Zhang, Y.; Qi, L.; Li, X.; Wang, W.; Li, C.; Li, X.; and Yang, M.-H. 2025. Controllable 3D outdoor scene generation via scene graphs. *arXiv preprint arXiv:2503.07152*.

Lugmayr, A.; Danelljan, M.; Romero, A.; Yu, F.; Timofte, R.; and Van Gool, L. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11461–11471.

Ma, X.; Qin, C.; You, H.; Ran, H.; and Fu, Y. 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv preprint arXiv:2202.07123*.

Mei, J.; Hu, T.; Yang, X.; Wen, L.; Yang, Y.; Wei, T.; Ma, Y.; Dou, M.; Shi, B.; and Liu, Y. 2024. DreamForge: Motion-aware autoregressive video generation for multi-view driving scenes. *arXiv preprint arXiv:2409.04003*.

Nakashima, K.; and Kurazume, R. 2024. LiDAR data synthesis with denoising diffusion probabilistic models. In *IEEE International Conference on Robotics and Automation*, 14724–14731.

Ni, J.; Guo, Y.; Liu, Y.; Chen, R.; Lu, L.; and Wu, Z. 2025. OpenDWM: Open Driving World Models. Https://github.com/SenseTime-FVG/OpenDWM.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, 4195–4205.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763. PmLR.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241. Springer.

Swerdlow, A.; Xu, R.; and Zhou, B. 2024. Street-view image generation from a bird's-eye view layout. *IEEE Robotics and Automation Letters*, 9(4): 3578–3585.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, L.; Zheng, W.; Ren, Y.; Jiang, H.; Cui, Z.; Yu, H.; and Lu, J. 2024. OccSora: 4D occupancy generation models as world simulators for autonomous driving. *arXiv preprint arXiv:2405.20337*.

Wu, Y.; Zhang, K.; Qian, J.; Xie, J.; and Yang, J. 2024. Text2LiDAR: Text-guided LiDAR point cloud generation via equirectangular transformer. In *European Conference on Computer Vision*, 291–310. Springer.

Xiong, Y.; Ma, W.-C.; Wang, J.; and Urtasun, R. 2023. UltraLiDAR: Learning compact representations for LiDAR completion and generation. *arXiv preprint arXiv:2311.01448*.

Xu, X.; Kong, L.; Shuai, H.; and Liu, Q. 2025. FRNet: Frustum-range networks for scalable LiDAR segmentation. *IEEE Transactions on Image Processing*, 34: 2173–2186.

Yan, T.; Yin, J.; Lang, X.; Yang, R.; Xu, C.-Z.; and Shen, J. 2025. OLiDM: Object-aware LiDAR diffusion models for autonomous driving. In *AAAI Conference on Artificial Intelligence*, 9121–9129.

Yang, X.; Wen, L.; Ma, Y.; Mei, J.; Li, X.; Wei, T.; Lei, W.; Fu, D.; Cai, P.; Dou, M.; et al. 2024. DriveArena: A closed-loop generative simulation platform for autonomous driving. *arXiv preprint arXiv:2408.00415*.

Yang, Y.; Mei, J.; Ma, Y.; Du, S.; Chen, W.; Qian, Y.; Feng, Y.; and Liu, Y. 2025a. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 9327–9335.

Yang, Z.; Lu, K.; Zhang, C.; Qi, J.; Jiang, H.; Ma, R.; Yin, S.; Xu, Y.; Xing, M.; Xiao, Z.; et al. 2025b. MMGDreamer: Mixed-modality graph for geometry-controllable 3D indoor scene generation. In *AAAI Conference on Artificial Intelligence*, 9391–9399.

Zhai, G.; Örnek, E. P.; Chen, D. Z.; Liao, R.; Di, Y.; Navab, N.; Tombari, F.; and Busam, B. 2024. EchoScene: Indoor scene generation via information echo over scene graph diffusion. In *European Conference on Computer Vision*, 167–184. Springer.

Zhang, L.; Xiong, Y.; Yang, Z.; Casas, S.; Hu, R.; and Urtasun, R. 2023. Copilot4D: Learning unsupervised world models for autonomous driving via discrete diffusion. *arXiv preprint arXiv:2311.01017*.

Zheng, W.; Chen, W.; Huang, Y.; Zhang, B.; Duan, Y.; and Lu, J. 2024a. OccWorld: Learning a 3D occupancy world model for autonomous driving. In *European Conference on Computer Vision*, 55–72. Springer.

Zheng, X.; Huang, X.; Mei, G.; Hou, Y.; Lyu, Z.; Dai, B.; Ouyang, W.; and Gong, Y. 2024b. Point cloud pre-training with diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22935–22945.

Zyrianov, V.; Zhu, X.; and Wang, S. 2022. Learning to generate realistic LiDAR point clouds. In *European Conference on Computer Vision*, 17–35. Springer.