MoE4D: Mixture of Spatio-Temporal Experts with Frequency-Modulated Router for LiDAR Novel View Synthesis

Anonymous authors
Paper under double-blind review

Abstract

Although recent progress in 4D scene representations significantly advances LiDAR novel view synthesis, it still remains inconclusive which spatio-temporal representation performs better across diverse regions and time-variant motion patterns. NeRF-based methods encode continuous 4D implicit field with superior parameter efficiency, while struggling in long-term dynamics and occlusion scenarios. In contrast, Gaussian Splatting-based methods enable fast real-time rendering but commonly require complicated ray-object intersection modeling, dense trajectory labels of dynamics, and explicit storage of Gaussian primitives. In this paper, we design a novel differential LiDAR simulation method based on Mixture-of-Experts, called MoE4D, to effectively combine various spatio-temporal cues within a unified rendering pipeline. Specifically, multi-planar, hash grid, and object-centric GS serve as main expert candidates, providing multi-granularity geometry and deformation priors. We also observe that previous works mostly overlook spectral discrepancies within LiDAR sequences. To mitigate this issue, a frequency-modulated context router is designed based on the Fourier analysis of input 4D signals. Furthermore, a dense-to-sparse expert distillation strategy is developed to maintain high-fidelity reconstruction quality with significantly reduced computational budgets. Extensive experiments on KITTI-360 and nuScenes demonstrate state-of-the-art performance of our MoE4D. Codes will be released upon publication.

1 Introduction

LiDAR novel view synthesis (LNVS) aims to reconstruct unseen frames from novel spatio-temporal view given a sequence of observed LiDAR sequences, which can create corner cases, improve data diversity, and facilitate closed-loop resimulation in autonomous driving and robotic domains (Zhou et al., 2025; Tao et al., 2024). Although camera-based reconstruction has been widely explored, LiDAR-based counterpart remains an open issue. Challenges in the LNVS task arise mainly from intrinsic sparsity, temporal inconsistency, large motion, and occlusions within the LiDAR data (Jiang et al., 2025).

To achieve high-fidelity LiDAR simulation, traditional LNVS approaches (Manivasagam et al., 2020; Li et al., 2022) explicitly reconstruct surfaces with mesh or point cloud, and calculate the ray-surface intersection to render novel LiDAR points. However, these methods are mostly limited to static scenarios and also sensitive to surface reconstruction quality (Zheng et al., 2024).

With great advancements in neural radiance fields (NeRF) (Mildenhall et al., 2021), recent LiDAR NeRFs (Tao et al., 2024; Zhang et al., 2024) implicitly represent continuous time-variant space and synthesize novel views via differential volume rendering. Subsequent followers strengthen frame-level temporal consistency with pretrained scene flow estimators (Liu et al., 2025; Yu et al., 2025), impose spatial geometry constraint (Yu et al.), or simultaneously optimize pose (Xue et al., 2024). However, LiDAR NeRFs suffer from rendering inefficiency due to the numerous ray sampling, and also face challenges in long-distance vehicle motion and occlusions (Jiang et al., 2025). To address these problems, some recent works investigate Gaussian Splatting-based LiDAR reconstruction with boosted rendering speed (Zhou et al., 2025; Jiang et al., 2025). However, most of them heavily rely

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

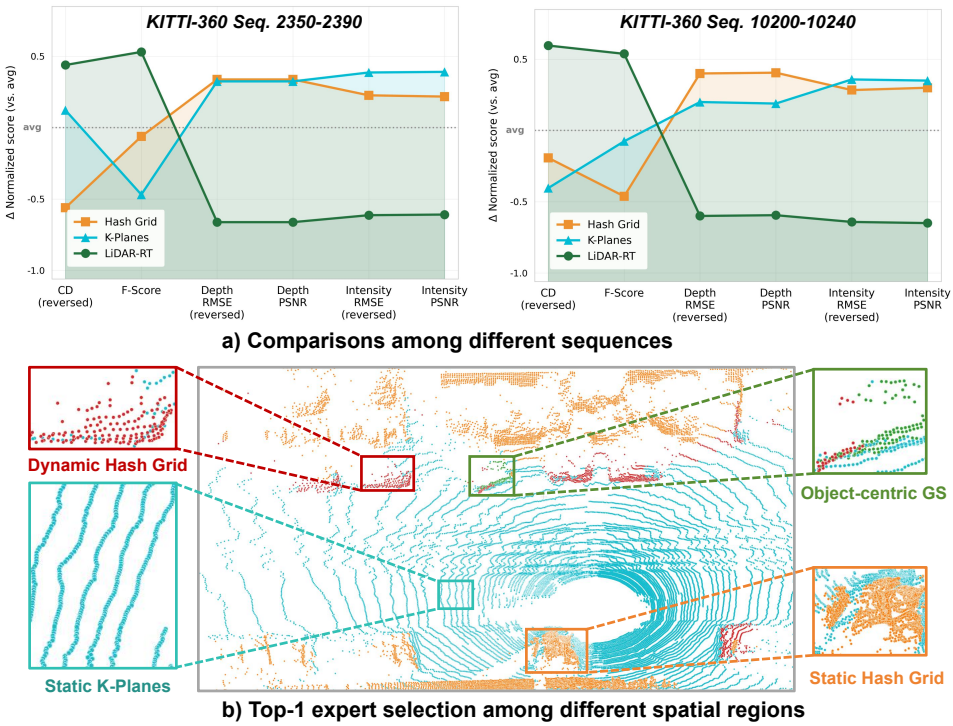


Figure 1: **Complementary strengths of spatio-temporal representations across different evaluation sequences and spatial regions.** a) Different experts excel on different metrics, and their relative rankings can vary greatly across different sequences. For example, K-Planes has better CD metric and worse F-Score on the Seq.2350-2390, compared to Hash grid. However, the relative strengths are reversed on the Seq.10200-10240. b) Within the same frame, different representations excel in various regions. Hash grid reconstructs better high-frequency details, while K-Planes performs well in low-frequency overall smoothness. Object-centric GS is designed for dynamics.

on complex ray-object intersection modeling and introduce huge storage costs of Gaussian primitives, especially in long-duration scenarios. Therefore, *it still remains unclear which representation performs better across diverse regions and motion patterns.*

As illustrated in Fig. 1, we observe that various 4D representations can provide complementary information and their strengths vary significantly across different sequences and even spatial regions. Motivated by this insight, we design a novel differential LiDAR NVS pipeline based on a mixture of spatio-temporal experts. Specifically, five expert candidates including multi-planar static, multi-planar dynamic, hash-grid static, hash-grid dynamic, and object centric Gaussian deformation field are introduced to combine complementary geometry and dynamic priors. In addition, spectral information naturally within LiDAR sequences has been largely overlooked in previous works, where low-frequency spectrum corresponds to smooth planes or coarse geometry structures and high-frequency spectrum captures geometric details or rapidly-moving dynamics. To effectively incorporate frequency awareness into our expert allocation, we develop a frequency-modulated context router based on the Fourier analysis of original 4D input, combining light-weight spatio-temporal context with spectral embeddings. To unleash the sparsity potential of MoE architecture, we also design a dense-to-sparse expert distillation strategy in our sparse routing mode, combining advantages in sufficient reconstruction priors from dense experts with high-efficiency top-k selection.

Overall, the key contributions of our paper are as follows:

- We propose a novel LiDAR novel view synthesis network based on Mixture of Spatio-temporal Experts, named MoE4D. Geometry and deformation priors from diverse 4D experts can be effectively combined to strengthen reconstruction quality.
- To incorporate spectrum-induced discrepancies, a frequency-modulated context router is designed, which combines light-weight spatio-temporal cues with spectral descriptors.

- A dense-to-sparse distillation strategy is developed to combine both sufficient reconstruction priors from dense 4D expert candidates and efficiency advantage in sparse routing.
- Our MoE4D outperforms previous methods across synthesis metrics in depth, intensity, and reconstructed points on KITTI-360 and nuScenes datasets. Furthermore, due to our designed distillation strategy, the sparse routing mode can largely boost rendering speed, retaining competitive performance with dense routing.

2 Related Work

2.1 LiDAR Novel View Synthesis

NeRF-based LiDAR NVS. Motivated by recent success of camera-based NeRFs (Mildenhall et al., 2021; Pumarola et al., 2021), LiDAR NeRFs have also been widely investigated. NeRF-LiDAR (Zhang et al., 2024) and UniSim (Yang et al., 2023) reconstruct urban scenes with both point cloud and image pairs. LiDAR-NeRF (Tao et al., 2024) and NFL (Huang et al., 2023) only require LiDAR points as supervision. Nevertheless, these models are commonly constrained by static scenes. To represent dynamics, LiDAR4D (Zheng et al., 2024) introduces scene flow representation as the deformation field. However, its temporal correlation is largely restricted in a local window. Although following approaches (Yu et al., 2025; Yu et al.; Liu et al., 2025) enlarge the temporal receptive field by introducing pre-trained flow estimators or history motion buffer, LiDAR NeRFs still suffer from slow rendering problem (Jiang et al., 2025).

GS-based LiDAR NVS. By explicitly reconstructing Gaussian primitives, recent Gaussian Splatting-based LiDAR reconstruction methods (Chen et al., 2026) achieve high-fidelity synthesis quality and boosted rendering speed, compared to LiDAR NeRFs. LiDAR-GS (Chen et al., 2024) incorporates view-dependent neural Gaussian fields for accurate surface splatting. LiDAR-RT (Zhou et al., 2025) designs Gaussian-based ray tracing against a bounding volume hierarchy to determine intersections. GS-LiDAR (Jiang et al., 2025) also computes ray-splat intersections and further employs 2D Gaussian primitives with periodic vibration properties. Despite obvious advantages in efficiency, GS-based LiDAR NVS methods mostly require precise ray-GS intersection calculation.

2.2 Mixture-of-Experts

The Mixture of Experts (MoE) technique enables flexible selection of multiple expert candidates, each of which specializes in a subtask. Within the standard MoE architecture, a gating network is commonly used to compute routing scores, which determine expert selection per input instance. Through a sparse routing strategy (Shazeer et al., 2017), recent MoE networks induce enhanced efficiency and scalability to large language models (LLMs) (Lepikhin et al., 2021; Yang et al., 2026) and computer vision tasks (Riquelme et al., 2021; Gao et al., 2026; Cai et al., 2025). Inspired by recent success in MoE-based image reconstruction (Jin et al., 2026b;a), we integrate multiple heterogeneous spatio-temporal experts, adaptively learning deformation and frequency priors.

3 MoE4D

3.1 Preliminaries

Problem Formulation. Given a consecutive sequence of LiDAR frames $\mathcal{P} = \{P_t\}_{t=1}^{M}$, along with its corresponding sensor pose $\mathcal{K} = \{K_t \in SE(3)\}_{t=1}^{M}$ and sampling timestamp t, the objective of LiDAR novel view synthesis is to render realistic LiDAR points from arbitrary new viewpoint K_{new} at timestamp t_{new}. $P_t \in \mathbb{R}^{N\times 4}$ indicates the N input points at frame t, consisting of both 3D coordinates and reflection intensity, and M denotes the total frame length.

Overall Pipeline. As in Fig. 2, our MoE4D queries complementary spatio-temporal representations on sampled points with five heterogeneous experts: static/dynamic multi-planar fields, static/dynamic hash-grid fields, and an object-centric Gaussian field. A lightweight frequency-modulated context router is designed to effectively predict expert preferences based on temporal and spectral cues. To achieve unified and efficient expert fusion, we select per-sample top-k repre-

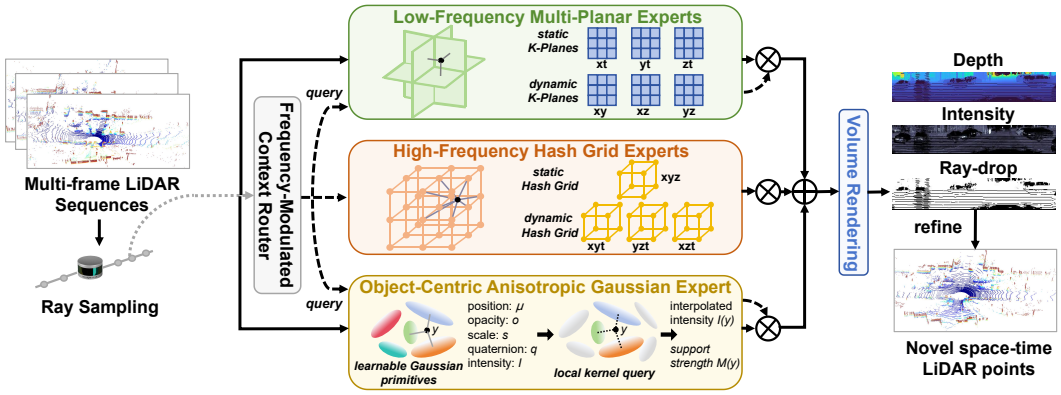


Figure 2: **Overall pipeline of our proposed MoE4D.** A frequency-modulated router dispatches each input sampling point into top-k available spatio-temporal experts. Afterwards, a unified volume rendering backend generates LiDAR attributes.

sentations before querying their encoded features. The queried features are finally gated and merged into a unified volume rendering process, generating LiDAR attributes.

3.2 Frequency-Modulated Context Router

We first introduce our designed MoE router which combines both light-weight spatio-temporal features and also spectral information to provide sufficient gating priors for expert allocation.

Lightweight Spatial and Temporal Routing Context. For each normalized query position X, we obtain local spatial features by trilinearly interpolating two learnable feature grids G_1 and G_2:

$$\mathbf{g}(X) = \mathrm{Concat}_{l=1}^{2}[\mathrm{GridSample}\,(G_l, X)], \tag{1}$$

where the resolutions of two grids are g_1 and g_2, respectively. Temporal embeddings are obtained by incorporating both the query time and sequence-level statistics as: $\mathbf{e}(X,t) = (\mathrm{PE}(X,t), \mathbf{v}(X,t))$, where $\mathrm{PE}(\cdot)$ indicates sinusoidal positional encoding of original 4D signal. $\mathbf{v}(X,t)$ is constructed by voxelizing valid LiDAR returns and summarizing temporal variations across all observed frames. Finally, the spatio-temporal context is: $\mathbf{c}(X,t) = \mathrm{Concat}[\mathbf{g}(X), \mathbf{e}(X,t)]$.

Fourier-based Spectral Descriptor. Previous methods commonly treat each spatio-temporal point equally, failing to distinguish high-frequency regions capturing geometrical details or fast motion and low-frequency regions such as smooth ground plane. To empower our routing network with spectral awareness, we further decompose the original 4D input based on Fourier analysis Tancik et al. (2020). Given spatial context $\mathbf{g}$, a lightweight coefficient prediction head produces B sine/cosine coefficient pairs $\{\mathbf{a}_b(X), \mathbf{b}_b(X)\}_{b=1}^{B}$ and a direct-current (DC, zero-frequency) coefficient $\mathbf{a}_0(X)$, each in $\mathbb{R}^{d_f}$. For positive frequencies $\nu_1 < \cdots < \nu_B$, we synthesize:

$$\mathbf{u}(X,t) = \mathbf{a}_0(X) + \sum_{b=1}^{B} \Big[\mathbf{a}_b(X)\cos(2\pi\nu_b t) + \mathbf{b}_b(X)\sin(2\pi\nu_b t)\Big]. \tag{2}$$

The band powers are then computed by: $P_0 = \|\mathbf{a}_0\|_2^2/d_f$, $P_b = \left(\|\mathbf{a}_b\|_2^2 + \|\mathbf{b}_b\|_2^2\right)/2d_f$. Let $Z = \sum_{b=0}^{B} P_b$, $\pi_b = P_b/Z$. Finally, the output spectral descriptor $\mathbf{d}_{\mathrm{freq}}$ concatenates normalized power π_b, spectral centroid μ_ν, spectral entropy H_ν, and total power Z as: $\mathbf{d}_{\mathrm{freq}} = \mathrm{Concat}[\pi_0, \ldots, \pi_B, \mu_\nu, H_\nu, \log(1+Z)]$, where $\mu_\nu = \sum_{b=0}^{B} \pi_b \frac{\nu_b}{\nu_B}$, $H_\nu = -\frac{\sum_{b=0}^{B} \pi_b \log(\pi_b+\epsilon)}{\log(B+1)}$.

FiLM-based Frequency Modulation. We follow the feature-wise affine conditioning in FiLM (Perez et al., 2018) to modulate spatio-temporal routing context with frequency descriptor:

$$[\boldsymbol{\gamma}, \boldsymbol{\beta}] = W_f \mathbf{d}_{\mathrm{freq}} + \mathbf{b}_f, \quad \widetilde{\mathbf{c}} = \mathbf{c} \odot [1 + 0.5\tanh(\boldsymbol{\gamma})] + \boldsymbol{\beta}. \tag{3}$$

The frequency branch controls both channel scaling and offsets before expert scoring. We initialize W_f and $\mathbf{b}_f$ to zero, yielding identity modulation at initialization. The final frequency-modulated descriptor $\widetilde{\mathbf{c}}$ combines both spatio-temporal cues and spectral information for expert allocation.

3.3 MIXTURE OF SPATIO-TEMPORAL EXPERTS

To integrate complementary modeling biases, our expert pool separates 4D representations with different resolutions and deformation priors. Specifically, we select multi-planar and hash-grid NeRFs, specializing in low-frequency and high-frequency regions, respectively (Zheng et al., 2024). However, NeRF-based methods struggle with long-distance motions (Yu et al., 2025). To address this problem, we supplement an object-centric Gaussian to track instance-level dynamics. Finally, the features from heterogeneous experts are gated and fused together to decode the LiDAR attributes.

Low-Frequency Multi-planar Experts. We adopt K-Planes (Fridovich-Keil et al., 2023) at lower resolutions to extract low-frequency features for overall smoothness. The input 4D coordinates ($X = \{xyz\}, t$) are decomposed into three spatial orthogonal planes (xy, xz, yz) and three space-time planes (xt, yt, zt). Static expert $(\mathcal{L}_{xy} \odot \mathcal{L}_{xz} \odot \mathcal{L}_{yz})$, and dynamic expert $(\mathcal{L}_{xt} \odot \mathcal{L}_{yt} \odot \mathcal{L}_{zt})$ are independently queried, bilinearly interpolated, and multiplied by Hadamard product.

High-Frequency Hash Grid Experts. We employ Instant-NGP (Müller et al., 2022) to instantiate static $\mathcal{H}^l_{xyz}$ and dynamic $(\mathcal{H}^l_{xyt} \odot \mathcal{H}^l_{xzt} \odot \mathcal{H}^l_{yzt})$ experts. We set hash-grid experts with much finer-grained resolutions, specializing in high-frequency geometric details and subtle dynamic variations. Features are stored in multi-resolution hash tables and queried at the projected coordinates.

Object-Centric Anisotropic Gaussian Expert. We design an object-centric anisotropic Gaussian feature field. Inspired by LiDAR-RT (Zhou et al., 2025), we also additionally input per-frame bounding boxes $\mathcal{B}$ depicting tracked motion for each dynamic instance. The Gaussian expert is available only within active object boxes for foreground dynamics. Each learnable Gaussian primitive j includes canonical center $\boldsymbol{\mu}_j$, scale s_j, orientation q_j, an opacity-like kernel amplitude o_j, and intensity-initialized learnable reflectance attribute c_j. For a query point $\mathbf{y}$ in the object's canonical coordinates, we compute the truncated anisotropic kernel as: $\kappa_j(\mathbf{y}) = o_j \exp\left(-\frac{1}{2} d_j^2(\mathbf{y})\right) \mathbb{1}\left[d_j^2(\mathbf{y}) \le r_G^2\right]$, where $d_j^2(\mathbf{y}) = (\mathbf{y} - \boldsymbol{\mu}_j)^\top \Sigma_j^{-1} (\mathbf{y} - \boldsymbol{\mu}_j)$. For each dynamic object, we retain up to J kernels with the largest responses, denoted by $\mathcal{N}_J(\mathbf{y})$, and aggregate:

$$M(\mathbf{y}) = \sum_{j \in \mathcal{N}_J(\mathbf{y})} \kappa_j(\mathbf{y}), \quad \bar{c}(\mathbf{y}) = \frac{\sum_{j \in \mathcal{N}_J(\mathbf{y})} \kappa_j(\mathbf{y}) c_j}{\max(M(\mathbf{y}), \epsilon)}, \quad \mathbf{f}_G(\mathbf{y}) = [\bar{c}(\mathbf{y}), \log(1 + M(\mathbf{y}))]\,. \quad (4)$$

The first attribute $\bar{c}(\mathbf{y})$ provides a local reflectance cue, while the second feature $\log(1 + M(\mathbf{y}))$ preserves the Gaussian support strength. Queries without kernel support produce zero features.

Gating and Expert Selection. We combine experts through a shared per-sample gating module. For i-th query sample, routing context $\widetilde{\mathbf{c}}_i$ is first mapped to per-expert logits as: ℓ_i^e, where e denotes the expert index. Then, logits are normalized over available experts: $p_i^e = \frac{A_i^e \exp(\ell_i^e)}{\sum_e A_i^e \exp(\ell_i^e)}$, where A_i^e=$\{0,1\}$ indicates whether the expert e is available for sample i. Given an expert budget k_i, we select the top-k_i experts with the highest probabilities as $\mathcal{E}_i$. Routing precedes among all ray samples, and only the selected experts are activated to compute their features $\mathbf{f}_i^e$ for efficiency. Finally, we fuse the selected representations as: $\mathbf{F}_i = \sum_{e \in \mathcal{E}_i} w_i^e \left(\mathbf{f}_i^e + \lambda_M \,\mathrm{MLP}_e(\mathbf{f}_i^e)\right) + \mathbf{c}_i$, where MLP_e is an expert-specific MLP layer for feature residual, and λ_M is a learnable residual scale. The output feature $\mathbf{F}_i$ is then passed to the volume rendering network.

Unified LiDAR Rendering Backend. We follow LiDAR4D (Zheng et al., 2024) to reconstruct LiDAR points with a unified differential volume rendering pipeline. For each ray r, the MoE-aggregated features $\mathbf{F}_i$ of all samples along the ray ($i \in r$) are fused into a density MLP to regress the density value and density features. Combining the density features with view embeddings, another two separate MLPs regress view-dependent intensity and ray-drop probability, respectively.

3.4 DISTILLATION-GUIDED ADAPTIVE SPARSE ROUTING

Apart from integrating heterogeneous priors, another intrinsic advantage of MoE stems from the sparsity, which significantly reduces computational costs compared to dense feature merging. However, intuitively decreasing the number of experts commonly leads to substantial performance degradation. To maintain high synthesis fidelity and also efficiency, we develop a dense-to-sparse distillation module, which learns informative expert-selection preferences from pre-trained dense teachers.

Table 1: **Quantitative comparisons on the KITTI-360 dataset**. The best and second-best results are highlighted. We classify existing methods into four categories: traditional simulator (Tra.), NeRF-based, GS-based, and hybrid (H.).

	Method	Point Cloud		Depth					Intensity				
		CD↓	F-score↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑
Tra.	LiDARsim	3.2228	0.7157	6.9153	0.1279	0.2926	0.6342	21.4608	0.1666	0.0569	0.3276	0.3502	15.5853
	NKSR	1.8982	0.6855	5.8403	0.0996	0.2752	0.6409	23.0368	0.1742	0.0590	0.3337	0.3517	15.2081
	PCGen	0.4636	0.8023	5.6583	0.2040	0.5391	0.4903	23.1675	0.1970	0.0763	0.5926	0.1351	14.1181
NeRF	LiDAR-NeRF	0.1438	0.9091	4.1753	0.0566	0.2797	0.6568	25.9878	0.1404	0.0443	0.3135	0.3831	17.1549
	D-NeRF	0.1442	0.9128	4.0194	0.0508	0.3061	0.6634	26.2344	0.1369	0.0440	0.3409	0.3748	17.3554
	TiNeuVox-B	0.1748	0.9059	4.1284	0.0502	0.3427	0.6514	26.0267	0.1363	0.0453	0.4365	0.3457	17.3535
	K-Planes	0.1302	0.9123	4.1322	0.0539	0.3457	0.6385	26.0236	0.1415	0.0498	0.4081	0.3008	17.0167
	LiDAR4D	0.1089	0.9272	3.5256	0.0404	0.1051	0.7647	27.4767	0.1195	0.0327	0.1845	0.5304	18.5561
	DifFlow3D	0.1070	0.9319	3.0786	0.0280	0.0701	0.8761	28.6807	0.0995	0.0263	0.1514	0.6561	20.0921
	STGC-NERF	0.0997	0.9325	3.0794	0.0277	0.0681	0.8774	28.6796	0.0995	0.0262	0.1479	0.6563	20.0825
GS	LiDAR-GS	0.1288	0.9156	3.9333	0.0530	0.2475	0.6276	26.1326	0.1336	0.0453	0.3411	0.3737	17.4554
	GS-LiDAR	0.1085	0.9231	3.1212	0.0340	0.0902	0.8553	28.4381	0.1161	0.0313	0.1825	0.5914	18.7482
	LiDAR-RT	0.1077	0.9255	3.4671	0.0512	0.1016	0.8406	27.6755	0.1115	0.0271	0.1812	0.6077	19.0862
H.	**MoE4D**	**0.0947**	**0.9391**	**3.0720**	**0.0249**	**0.0704**	**0.8780**	**28.7496**	**0.0989**	**0.0261**	**0.1468**	**0.6603**	**20.1314**

Dense-to-Sparse Distillation. We transfer expert priors from a higher-budget teacher to an adaptive lower-budget student, aiming to preserve reconstruction quality with fewer expert reservations. The student with budget k_s is initialized from a pretrained fixed top-k_d routing teacher network. For the i-th query point, let $\mathbf{F}_i^S, \mathbf{F}_i^T \in \mathbb{R}^D$ denote the fused features before density decoder in student and teacher networks, respectively, σ_i^S, σ_i^T denote their predicted densities. To retain both representation and geometry consistency, we match features and densities over Q uniformly sampled queries as:

$$\mathcal{L}_{\text{KD}} = \frac{\frac{1}{QD}\sum_{i=1}^{Q}\left\|\mathbf{F}_i^S - \mathbf{F}_i^T\right\|_2^2}{\max\left(\frac{1}{QD}\sum_{i=1}^{Q}\left\|\mathbf{F}_i^T\right\|_2^2, \epsilon_F\right)} + \frac{1}{Q}\sum_{i=1}^{Q}\ell_{\text{SL1}}\left(\log(1+\sigma_i^S), \log(1+\sigma_i^T)\right), \tag{5}$$

where ℓ_{SL1} is the Smooth L1 loss. Teacher targets are detached throughout student training.

Ambiguity-guided Adaptive Routing. After distillation, we adaptively allocate either k_s or k_d experts to each query point, according to per-point reconstruction ambiguity. Let $p_{i,(z)}$ denote the z-th largest probability among all available experts. We quantify the probability mass omitted by sparse routing: $d_i = 1 - \sum_{z=1}^{k_s} p_{i,(z)}$. A larger d_i indicates that the top-k_s sparse experts fail to provide sufficient reconstruction priors, motivating the network to involve additional experts. Thus, for each ray r with N_r samples, we select the $\lfloor \rho N_r \rfloor$ samples with the largest ambiguity scores and leverage k_d experts. Overall, the average budget of approximately $k_s + \rho(k_d - k_s)$. Inference uses the same adaptive allocation rule and requires only the student network.

3.5 Loss Functions

We combine reconstruction loss functions following (Yu et al.; Zheng et al., 2024; Yu et al., 2025), including depth loss, intensity loss, and ray-drop loss:

$$\mathcal{L}_{\text{d}} = \sum_{\mathbf{r}\in\mathcal{R}}\left\|\widehat{D}(\mathbf{r}) - D(\mathbf{r})\right\|_1, \mathcal{L}_{\text{in}} = \sum_{\mathbf{r}\in\mathcal{R}}\left\|\widehat{I}(\mathbf{r}) - I(\mathbf{r})\right\|_2^2, \mathcal{L}_{\text{ray}} = \sum_{\mathbf{r}\in\mathcal{R}}\left\|\widehat{P}(\mathbf{r}) - P(\mathbf{r})\right\|_2^2, \tag{6}$$

where $\mathbf{r} \in \mathcal{R}$ indicates the casted rays. Finally, the overall pipeline is optimized by:

$$\mathcal{L}_{all} = \lambda_d\mathcal{L}_{\text{d}} + \lambda_{in}\mathcal{L}_{\text{in}} + \lambda_{ray}\mathcal{L}_{\text{ray}} + \lambda_{KD}\mathcal{L}_{\text{KD}}. \tag{7}$$

4 Experiment

4.1 Experimental Settings

Datasets and Metrics. We evaluate MoE4D on two commonly-used autonomous driving datasets: KITTI-360 (Liao et al., 2022) and nuScenes (Caesar et al., 2020). We follow (Zheng et al., 2024) to curate training/testing sequences and assess LiDAR synthesis quality by Chamfer Distance (CD), F-Score, RMSE, MedAE, PSNR, SSIM, and LPIPS. Please refer to the appendix for more details.

Table 2: **Quantitative comparisons on the KITTI-360 *static* dataset.**

	Method	Point Cloud		Depth					Intensity				
		CD↓	F-score↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑
Tra.	LiDARsim	2.2249	0.8667	6.5470	0.0759	0.2289	0.7157	21.7746	0.1532	0.0506	0.2502	0.4479	16.3045
	NKSR	0.5780	0.8685	4.6647	0.0698	0.2295	0.7052	22.5390	0.1565	0.0536	0.2429	0.4200	16.1159
	PCGen	0.2090	0.8597	4.8838	0.1785	0.5210	0.5062	24.3050	0.2005	0.0818	0.6100	0.1248	13.9606
NeRF	LiDAR-NeRF	0.0923	0.9226	3.6801	0.0667	0.3523	0.6043	26.7663	0.1557	0.0549	0.4212	0.2768	16.1683
	LiDAR4D	0.0894	0.9264	3.2370	0.0507	0.1313	0.7218	27.8840	0.1343	0.0404	0.2127	0.4698	17.4529
	STGC-NeRF	0.0831	0.9332	2.7717	0.0353	0.0985	0.8480	29.2388	0.1121	0.0336	0.1822	0.6116	19.0238
GS	LiDAR-GS	0.0911	0.9232	3.3879	0.0606	0.2797	0.6228	26.9887	0.1380	0.0473	0.2725	0.4381	17.3459
	GS-LiDAR	0.0847	0.9236	2.8895	0.0411	0.0997	0.8454	28.8807	0.1211	0.0359	0.1630	0.5756	18.3506
	LiDAR-RT	0.0854	0.9216	2.8859	0.0467	0.1004	0.8455	29.0153	0.1195	0.0355	0.1808	0.5884	18.5903
H.	**MoE4D**	**0.0752**	**0.9390**	**2.7370**	**0.0331**	**0.0983**	**0.8501**	**29.3440**	**0.1113**	**0.0330**	**0.1769**	**0.6142**	**19.0860**

Table 3: **Quantitative comparisons on the nuScenes dataset.**

	Method	Point Cloud		Depth					Intensity				
		CD↓	F-score↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑
Tra.	LiDARsim	12.1383	0.6512	10.5539	0.3572	0.1871	0.5653	17.7841	0.0659	0.0115	0.1160	0.5170	23.7791
	NKSR	11.4910	0.6178	9.3731	0.5763	0.2111	0.5637	18.7774	0.0680	0.0119	0.1290	0.5031	23.4905
	PCGen	2.1998	0.6341	8.8364	0.4011	0.1792	0.5440	19.2799	0.0768	0.0147	0.1308	0.4410	22.4428
NeRF	LiDAR-NeRF	0.3225	0.8576	7.1566	0.0338	0.0702	0.7188	21.2129	0.0467	0.0076	0.0483	0.7264	26.9927
	D-NeRF	0.3296	0.8513	7.1089	0.0368	0.0789	0.7130	21.2594	0.0467	0.0080	0.0492	0.7180	26.9951
	TiNeuVox-B	0.3920	0.8627	7.2093	0.0290	0.1549	0.6873	21.0932	0.0462	0.0080	0.1294	0.7107	26.8620
	K-Planes	0.2982	0.8887	6.7960	0.0209	0.1218	0.7258	21.6203	0.0438	0.0076	0.1127	0.7364	27.4227
	LiDAR4D	0.2443	0.8915	6.7831	0.0258	0.0569	0.7396	21.7189	0.0426	0.0071	0.0459	0.7498	27.7977
	STGC-NeRF	0.2204	0.9070	6.5361	0.0240	0.0486	0.7741	22.0044	0.0417	0.0082	0.0418	0.7566	27.9989
GS	LiDAR-GS	0.2829	0.8987	6.8806	0.0290	0.1108	0.7286	21.7803	0.0444	0.0077	0.0527	0.7346	27.2427
	GS-LiDAR	0.2382	0.9055	5.8925	0.0198	0.0708	0.8394	22.2482	0.0415	0.0067	0.0627	0.7291	27.7420
	LiDAR-RT	0.2300	0.9000	6.6195	0.0245	0.0493	0.7655	21.8720	0.0422	0.0081	0.0483	0.7542	27.8105
H.	**MoE4D**	**0.1891**	**0.9246**	**6.5051**	**0.0134**	**0.0505**	**0.7816**	**21.9398**	**0.0380**	**0.0055**	**0.0371**	**0.7987**	**28.6864**

Baselines. We compare our model with (1) traditional simulators: LiDARsim (Manivasagam et al., 2020), NKSR (Huang et al., 2023), PCGen (Li et al., 2022); (2) NeRF-based methods: LiDAR-NeRF (Tao et al., 2024), LiDAR4D (Zheng et al., 2024), DifFlow3D (Liu et al., 2025), STGC-NeRF (Yu et al., 2025), MAC-NeRF (Yu et al.), D-NeRF (Pumarola et al., 2021), TiNeuVox-B (Fang et al., 2022), and K-Planes (Fridovich-Keil et al., 2023); (3) GS-based methods: LiDAR-GS (Chen et al., 2024), GS-LiDAR (Jiang et al., 2025), and LiDAR-RT (Zhou et al., 2025).

Implementation Details. Adhering to the protocol in (Yu et al., 2025), we re-scale the input point cloud within a unit cube. All experiments are conducted on a single NVIDIA RTX 4090 GPU. The model is optimized by Adam optimizer for 30K iterations per scene with a batch size of 1024 rays, followed by a ray-drop refinement stage (Zheng et al., 2024). The initial learning rate is set as 1×10^{-2}. Along each ray, N_r=768 points are uniformly sampled. g_1 and g_2 are set as 16 and 32, respectively. For frequency coefficients, we set $\mathbf{B}$ as 3 and (ν_1, ν_2, ν_3)=(1,2,3). In GS expert, we set J as 8 with the largest truncated Mahalanobis responses. For distillation, $Q = 4096$, $k_s = 2$, $k_d = 4$ and $\rho = 10\%$ as default. $(\lambda_d, \lambda_{in}, \lambda_{ray}, \lambda_{KD}) = (1, 0.1, 0.01, 0.001)$.

4.2 Quantitative Comparison

Comparison Results on KITTI-360 Dataset. As in Table 1, our MoE4D outperforms all existing traditional, NeRF-based, and GS-based methods in almost all metrics on KITTI-360 dynamic sequences. Specifically, our model achieves 0.0947 point CD, 0.9391 F-Score, 3.0720 depth RMSE, and 0.0989 intensity RMSE. Although there is only 0.0023 higher depth LPIPS compared to the recent state-of-the-art method STGC-NeRF, we still gain much better metrics for all other depth metrics. Furthermore, we also report the comparison results on static sequences in Table 2, where our MoE4D also gets significantly enhanced reconstruction quality in both 3D and 2D metrics. This demonstrates that the integrated geometry and deformation priors from heterogeneous experts can effectively improve LiDAR synthesis quality.

Comparison Results on nuScenes Dataset. We also evaluate our model on the nuScenes dataset in Table 3. Our MoE4D achieves the best reconstruction quality in generated point cloud, reducing the CD metric by 14.2%, compared to the second-best method STGC-NeRF. As for the projected range

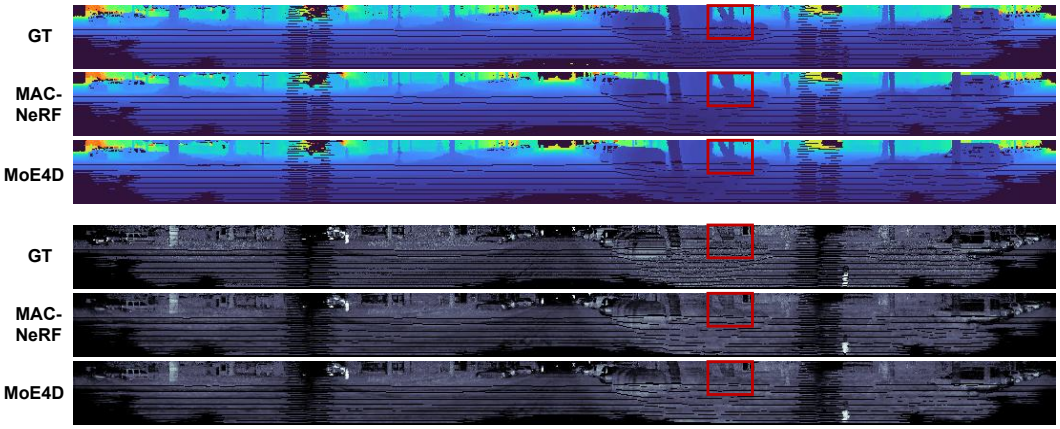


Figure 3: **Qualitative comparisons of the reconstructed depth and intensity.**

Table 4: **Ablation studies on the KITTI-360 dataset**. The best results are highlighted.

Settings	Point Cloud		Depth					Intensity				
	CD↓	F-score↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑
(a) Router												
w/o spatial grids	0.0970	0.9312	3.7253	0.0409	0.2486	0.7220	26.9493	0.1247	0.0340	0.2544	0.4897	18.1604
w/o temporal embeddings	0.1002	0.9307	3.6869	0.0410	0.2463	0.7270	27.0413	0.1238	0.0332	0.2500	0.4994	18.2197
w/o spectral descriptors	0.1002	0.9341	3.0790	0.0274	**0.0703**	0.8772	28.6977	0.0992	0.0265	0.1542	0.6572	20.1050
All three features	**0.0947**	**0.9391**	**3.0720**	**0.0249**	0.0704	**0.8780**	**28.7496**	**0.0989**	**0.0261**	**0.1468**	**0.6603**	**20.1314**
(b) Expert												
w/o GS expert	**0.0935**	0.9336	3.6120	0.0379	0.2308	0.7387	27.3035	0.1219	0.0332	0.2415	0.5073	18.3937
w/o Hash-grid Expert	0.1001	0.9291	3.0923	0.0285	0.0772	0.8776	28.6561	**0.0984**	0.0268	0.2428	0.6465	**20.1662**
w/o K-Planes Expert	0.1113	0.9240	3.0934	0.0316	0.0725	**0.8798**	28.6738	0.1001	0.0279	0.1523	0.6470	20.0281
All experts	0.0947	**0.9391**	**3.0720**	**0.0249**	**0.0704**	0.8780	**28.7496**	0.0989	**0.0261**	**0.1468**	**0.6603**	20.1314
(c) Sparse routing												
Top-1	0.1466	0.9055	3.2321	0.0371	0.0865	0.8716	28.3113	0.1031	0.0299	0.2320	0.6157	19.7691
Top-2	0.1107	0.9228	3.1399	0.0330	0.0742	0.8717	28.5236	0.1010	0.0277	0.1597	0.6429	19.9515
Top-3	0.0977	0.9307	3.1279	0.0309	0.0732	0.8769	28.5573	0.1008	0.0259	0.1499	0.6521	19.9729
Top-4	**0.0936**	0.9322	3.1157	0.0301	0.0725	0.8721	28.5590	0.0998	**0.0253**	0.1492	0.6592	20.0633
All experts	0.0947	**0.9391**	**3.0720**	**0.0249**	**0.0704**	**0.8780**	**28.7496**	**0.0989**	0.0261	**0.1468**	**0.6603**	20.1314

images, our method also achieves competitive performance in depth and intensity. This demonstrates that MoE4D unleashes great potential by integrating various 4D representations.

4.3 Qualitative Comparison

As in Fig. 3, we visualize the reconstructed depth and intensity on the KITTI-360 dataset. Compared to recent SoTA method MAC-NeRF, our method has better synthesis with sharper object boundaries. This is mainly because our network can combine complementary geometry-modeling strengths from various experts. We also compare the final reconstructed LiDAR points in Fig. 4. Compared to MAC-NeRF, our generated output aligns closer to GT points.

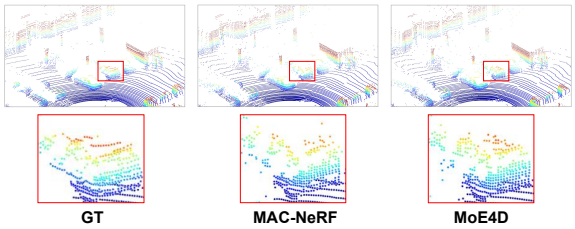


Figure 4: Visualization of reconstructed point cloud.

4.4 Ablation Studies

We conduct ablations on key module designs: router, expert, and sparse routing. As in Table 4 (a), removing each of the three routing features (spatial, temporal, spectral) leads to much worse performance. This proves that combining spatio-temporal cues with frequency information can provide richer gating proofs, which helps the router to select appropriate experts. Furthermore, integrating all five experts can produce the best overall reconstruction quality (Table 4 (b)). Even though removing GS expert can improve the reconstructed LiDAR quality slightly (0.0012 lower

Table 5: **Compatibility evaluation on various baselines.** Our MoE architecture naturally supports flexible incorporation of more advanced representations. The best results are highlighted.

Settings	Point Cloud		Depth					Intensity				
	CD↓	F-score↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑
LiDAR4D	0.1089	0.9272	3.5256	0.0404	0.1051	0.7647	27.4767	0.1195	0.0327	0.1845	0.5304	18.5561
LiDAR4D+Ours	**0.0947**	**0.9391**	**3.0720**	**0.0249**	**0.0704**	**0.8780**	**28.7496**	**0.0989**	**0.0261**	**0.1468**	**0.6603**	**20.1314**
Improv.	+13.04%	+1.28%	+12.87%	+38.37%	+33.02%	+14.82%	+4.63%	+17.24%	+20.18%	+20.43%	+24.49%	+8.49%
LiDAR-RT	0.1077	0.9255	3.4671	0.0512	0.1016	**0.8406**	27.6755	0.1115	**0.0271**	0.1812	0.6077	19.0862
LiDAR-RT+Ours	**0.0984**	**0.9312**	**3.2326**	**0.0342**	**0.1015**	0.8396	**28.2611**	**0.1052**	0.0274	**0.1698**	**0.6203**	**19.6665**
Improv.	+8.64%	+0.62%	+6.76%	+33.20%	+0.10%	-0.12%	+2.12%	+5.65%	-1.11%	+6.29%	+2.07%	+3.04%
MAC-NeRF	0.1044	0.9321	**3.0775**	0.0280	**0.0708**	0.8765	**28.6855**	0.0998	0.0263	0.1509	0.6553	20.0632
MAC-NeRF+Ours	**0.0951**	**0.9385**	3.0900	**0.0252**	0.0730	**0.8804**	28.6671	**0.0980**	**0.0254**	**0.1434**	**0.6698**	**20.2148**
Improv.	+8.91%	+0.69%	-0.41%	+10.00%	-3.11%	+0.44%	-0.06%	+1.80%	+3.42%	+4.97%	+2.21%	+0.76%

CD), there are obvious degradations on other metrics. This can be further proved by (c), where the fewer experts k_i leveraged, the worse LiDAR synthesis metrics are.

5 Discussions

Compatibility across Representation Backbones. To evaluate the compatibility of our MoE architecture, we integrate it into LiDAR4D, LiDAR-RT, and MAC-NeRF while retaining their original representations and rendering pipelines wherever possible. As in Table 5, our method consistently improves the point-cloud metrics across all three backbones and provides broad gains in intensity reconstruction. The improvement is most obvious for LiDAR4D, whereas the stronger baselines LiDAR-RT and MAC-NeRF exhibit smaller gains. We interpret this observation as reduced performance headroom: advanced backbones already capture part of the complementary spatio-temporal priors introduced by our expert mixture, leaving less room for additional representational capacity.

Distillation-Enabled Accuracy-Efficiency Trade-off. Table 6 further compares our distilled adaptive routing strategy with fixed top-k execution. Fixed routing provides only a limited number of discrete expert selection, and every query is assigned the same expert budget. In contrast, our dense-to-sparse distillation allocates the larger budget only to samples with high routing ambiguity. The allocation ratio ρ dynamically controls the expected number of active experts, thereby providing flexible accuracy-efficiency trade-off. Across the evaluated ratios, the distilled students require only 8.71-8.89 seconds per frame, which is close to the 7.78 seconds of fixed top-2 sparse routing and substantially lower than the 13.01 seconds of the higher-budget top-4 teacher. Meanwhile, all distilled variants outperform fixed top-3 routing across the reported point-cloud, depth, and intensity metrics while also reducing inference latency. These results show that distillation transfers most of the teacher's reconstruction priors to a student whose expert cost remains much closer to sparse routing, while ρ allows the averaged experts to be dynamically adjusted according to the available computational budget.

Table 6: **Comparison of between fixed routing and our distillation-guided adaptive routing.** The allocation ratio ρ flexibly controls the fraction of queries assigned the larger expert budget. Our distilled students retain near-teacher reconstruction quality while operating at a latency much closer to sparse routing.

Settings	Point Cloud		Depth RMSE↓	Intensity RMSE↓	Training time per step (s)	Latency per frame (s)
	CD↓	F-score↑				
Fixed (Top-2)	0.1107	0.9228	3.1399	0.1010	0.362	7.78
Fixed (Top-3)	0.0977	0.9307	3.1279	0.1008	0.482	10.82
Fixed (Top-4)	0.0936	0.9322	3.1157	0.0998	0.613	13.01
Distill(4→2, ρ=10%)	0.0970	0.9313	3.1001	0.1002	0.418	8.71
Distill(4→2, ρ=15%)	0.0975	0.9310	3.0988	0.1000	0.436	8.87
Distill(4→2, ρ=20%)	0.0962	0.9316	3.1243	0.1003	0.447	8.89

6 Conclusion

In this paper, we proposed a novel LiDAR novel view synthesis pipeline based on a mixture of heterogeneous spatio-temporal experts. A frequency-modulated context router is developed to incorporate sufficient spatio-temporal-spectral cues for gating logits. Furthermore, to improve reconstruction efficiency, we introduce a dense-to-sparse distillation network which can retain sufficient reconstruction quality comparable to large-budget teacher while requiring low computational costs closer to sparse routing. Extensive quantitative and qualitative comparisons with existing single-expert baselines demonstrate the superiority of our MoE-integrated representations.

AI USE STATEMENT

LLMs were used to assist with paper writing and presentation, including language polishing and checking formatting and textual consistency. It was not used to run experiments, generate experimental measurements, or determine the reported results. The authors are responsible for the final text, claims, figures, and results.

ETHICS STATEMENT

MoE4D is developed for research on LiDAR reconstruction and autonomous driving simulation. The proposed method generates novel-view spatio-temporal LiDAR points, but its performance in collected datasets does not establish safety concerns for real-world autonomous-driving deployment.

REPRODUCIBILITY STATEMENT

Section 3 presents the frequency-modulated router, mixture of spatio-temporal representation experts, and distillation-guided adaptive sparse routing. Section 4.1 reports the datasets, metrics, training configuration, inference settings, and detailed implementation details, while Table 4 provides ablation studies of the main design components. Appendix A.3 further details dataset and used metrics, and also initialization manners of object-centric GS.

REFERENCES

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 *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11618–11628. IEEE, 2020.

Wenrui Cai, Qingjie Liu, and Yunhong Wang. Spmtrack: Spatio-temporal parameter-efficient fine-tuning with mixture of experts for scalable visual tracking. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16871–16881. IEEE, 2025.

Qifeng Chen, Sheng Yang, Sicong Du, Tao Tang, Rengan Xie, Peng Chen, and Yuchi Huo. Lidar-gs: Real-time lidar re-simulation using gaussian splatting. *arXiv preprint arXiv:2410.05111*, 2024.

Qifeng Chen, Jiarun Liu, Rengan Xie, Tao Tang, Sicong Du, Yiru Zhao, Yuchi Huo, and Sheng Yang. Lidar-gs++: Improving lidar gaussian reconstruction via diffusion priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2975–2983, 2026.

Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 conference papers*, pp. 1–9, 2022.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12479–12488. IEEE, 2023.

Jingnan Gao, Zhe Wang, Xianze Fang, Xingyu Ren, Zhuo Chen, Shengqi Liu, Yuhao Cheng, Jiangjing Lyu, Xiaokang Yang, and Yichao Yan. More: 3d visual geometry reconstruction meets mixture-of-experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14680–14691, 2026.

Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4369–4379. IEEE, 2023.

Junzhe Jiang, Chun Gu, Yurui Chen, and Li Zhang. Gs-lidar: Generating realistic lidar point clouds with panoramic gaussian splatting. In *International Conference on Learning Representations*, volume 2025, pp. 56631–56645, 2025.

In-Hwan Jin, Hyeongju Mun, Joonsoo Kim, Kugjin Yun, and Kyeongbo Kong. On the design of mixture-of-experts for dynamic gaussian splatting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026a.

In-Hwan Jin, Hyeongju Mun, Joonsoo Kim, Kugjin Yun, and Kyeongbo Kong. Moe-gs: Mixture of experts for dynamic gaussian splatting. In *International Conference on Learning Representations*, volume 2026, pp. 62198–62220, 2026b.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

Chenqi Li, Yuan Ren, and Bingbing Liu. Pcgen: Point cloud generator for lidar simulation. *arXiv preprint arXiv:2210.08738*, 2022.

Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (3):3292–3310, 2022.

Jiuming Liu, Weicai Ye, Guangming Wang, Chaokang Jiang, Lei Pan, Jinru Han, Zhe Liu, Guofeng Zhang, and Hesheng Wang. Difflow3d: Hierarchical diffusion models for uncertainty-aware 3d scene flow estimation. *IEEE transactions on pattern analysis and machine intelligence*, 2025.

Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11164–11173. IEEE, 2020.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10313–10322. IEEE, 2021.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in neural information processing systems*, 34:8583–8595, 2021.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 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. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 390–398, 2024.

Weiyi Xue, Zehan Zheng, Fan Lu, Haiyun Wei, Guang Chen, and Changjun Jiang. Geonlf: Geometry guided pose-free neural lidar fields. *Advances in Neural Information Processing Systems*, 37: 73672–73692, 2024.

Haoqi Yang, Luohe Shi, Qiwei Li, Zuchao Li, Ping Wang, Hao Huang, and Hai Zhao. Faster moe llm inference for extremely large models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 43133–43151, 2026.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1389–1399. IEEE, 2023.

Shangshu Yu, Xiaotian Sun, Wen Li, Rui She, Hanyun Wang, Sheng Ao, Chenglu Wen, and Cheng Wang. Mac-nerf: Motion-aware curriculum learning for dynamic lidar nerfs. In *Forty-third International Conference on Machine Learning*.

Shangshu Yu, Xiaotian Sun, Wen Li, Qingshan Xu, Zhimin Yuan, Sijie Wang, Rui She, and Cheng Wang. Stgc-nerf: Spatial-temporal geometric consistency for lidar neural radiance fields in dynamic scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9644–9652, 2025.

Junge Zhang, Feihu Zhang, Shaochen Kuang, and Li Zhang. Nerf-lidar: Generating realistic lidar point clouds with neural radiance fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7178–7186, 2024.

Zehan Zheng, Fan Lu, Weiyi Xue, Guang Chen, and Changjun Jiang. Lidar4d: Dynamic neural fields for novel space-time view lidar synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5145–5154. IEEE, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1538–1548. IEEE, 2025.

A Appendix

A.1 Overview

In the appendix, we supplement more analysis and additional experimental results as follows:

- A limitation analysis is provided in Section A.2 to analyze the reason why previous single-expert network performs worse.
- Additional experimental settings are further discussed in Section A.3.
- More experimental results are given in Section A.4.

A.2 Limitation Analysis of Previous Works

NeRFs-based LiDAR NVS. LiDAR NeRFs (Tao et al., 2024; Zheng et al., 2024; Zhang et al., 2024) typically establish a continuous implicit representation based on neural radiance fields, mapping the input LiDAR points, view direction, and timestamp into a series of MLP layers. The output contains not only volume density but also LiDAR attributes such as intensity and ray-drop. Specifically, let $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ represent the LiDAR rays emitted from the sensor origin $\mathbf{o}$ in the direction $\mathbf{d}$, scene attributes including depth $D(\mathbf{r})$, intensity $I(\mathbf{r})$, and ray-drop $R(\mathbf{r})$ characteristics are sampled along the ray as:

$$D(\mathbf{r}) = \sum_{j=1}^{J} T_j(1 - e^{\sigma_j \delta_j}) d_j, \quad I(\mathbf{r}) = \sum_{j=1}^{J} T_j(1 - e^{\sigma_j \delta_j}) i_j, \quad R(\mathbf{r}) = \sum_{j=1}^{J} T_j(1 - e^{\sigma_j \delta_j}) r_j, \quad (8)$$

where $T_j = e^{-\sum_{s=1}^{j-1} \sigma_s \delta_s}$ indicates the accumulated transmittance, and δ_j is the distance between adjacent samples. d_j, i_j, r_j are the sampled depth, intensity, and ray-drop probability of the j-th sample, respectively.

GS-based LiDAR NVS. GS-based LiDAR simulators (Chen et al., 2024; 2026; Zhou et al., 2025) represent dynamic urban scenes based on explicit Gaussian primitives. GS-LiDAR (Jiang et al., 2025) employs 2D Gaussian primitives with periodic vibrations,and each primitive is assigned central point position μ, opacity o, scaling vector s and view-dependent intensity λ and ray-drop probability ρ modeled by spherical harmonics. Depth value can be obtained by computing the ray-splat intersection. Specifically, let r denote the distance from intersection point to LiDAR origin, the mean depth is:

$$D_{mean} = \sum_{h=1}^{H} r_h o_h \mathcal{G}_h \prod_{l=1}^{h-1} (1 - o_l \mathcal{G}_l), \tag{9}$$

where $\mathcal{G}_h$ denotes the influence of each Gaussian primitive. Intensity and ray-drop attributes are then rendered using alpha-blending as:

$$I = \sum_{h=1}^{H} \lambda_h o_h \mathcal{G}_h \prod_{l=1}^{h-1} (1 - o_l \mathcal{G}_l), \quad R = \sum_{h=1}^{H} \rho_h o_h \mathcal{G}_h \prod_{l=1}^{h-1} (1 - o_l \mathcal{G}_l). \tag{10}$$

Other works adopt similar Gaussian primitives and rendering strategy for LiDAR simulation.

Limitations of Single Representation. Compared to LiDAR NeRFs, GS-based methods significantly improve rendering efficiency. However, it relies on accurate surface-ray intersection and also fails to deal with long-duration sequences with increasingly accumulated primitives (Yu et al.). LiDAR NeRFs avoids explicit intersection compute while struggling with long-distance dynamics and occlusions. In this paper, we propose a unified LiDAR rendering network based on Mixure-of-Experts, adaptively combining advantages from multiple expert candidates.

A.3 Additional Experimental Settings

A.3.1 Datasets and Metrics

Datasets. KITTI-360 collects 64-beam LiDAR scans at a frequency of 10Hz, with 26.4 degree vertical FOV. nuScenes has sparser LiDAR points using 32-beam LiDAR sensor acquired at a 20-Hz frequency, with 40 degree vertical FOV. Following the standard protocol in (Tao et al., 2024; Zheng et al., 2024; Jiang et al., 2025; Yu et al.), we also comprehensively evaluate MoE4D on 6 dynamic sequences and 4 static sequences on KITTI-360, and 5 dynamic sequences on nuScenes. Dynamic sequences include 51 frames and every 10-th frame in the same interval is held-out for testing, while static sequences have 64 frames with 4 evenly sampled test frames.

Evaluation Metrics. We follow previous methods (Zheng et al., 2024; Chen et al., 2024; Zhou et al., 2025; Yu et al.), assessing both quality of 3D reconstructed points and also 2D projected range images. As for 3D metrics, Chamfer Distance (CD) quantifies geometry errors between generated and GT point cloud, while F-Sore is also reported with a 5-cm threshold. 2D metrics include Root Mean Square Error (RMSE) and Median Absolute Error (MedAE) to compute per-pixel averaged errors and also PSNR, SSIM, and LPIPS.

A.3.2 Initialization of Object-Centric GS

Centers and intensities of the GS expert are initialized from in-box LiDAR attributes in observed training frames. Scales are initialized from local coordinate-wise neighbor dispersion, orientations start at identity. Contributions from overlapping objects are combined by summing support masses and mass-weighting their intensities.

A.4 Additional Visualizations

We also supplement additional visualizations to compare our model with recent SoTA method MAC-NeRF. As shwon in Fig. 5, the depths reconstructed by our model are more similar to the GT ones on dynamic cars. This demonstrates that our designed MoE-based architecture can effectively combine different geometric and deformable priors using heterogeneous experts.

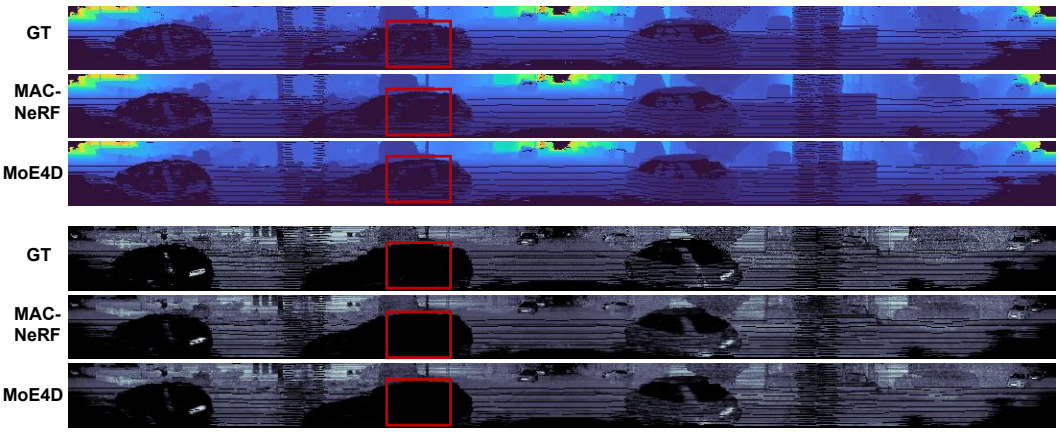


Figure 5: **Qualitative comparisons of the reconstructed depth and intensity.**