LiDAR-RT: Gaussian-based Ray Tracing for Dynamic LiDAR Re-simulation

Supplementary Material

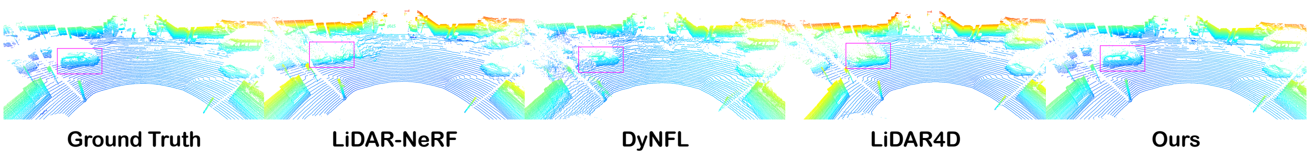

Figure 1. **Qualitative comparison of novel view LiDAR point clouds on Waymo Open Dataset [8].** Our LiDAR-RT generates realistic novel LiDAR views with accurate scene geometry and high-frequency details of dynamic objects.

In this supplementary material, we begin by presenting additional implementation details about the evaluation datasets, baseline methods and other technical descriptions in Sec. 1. Then we display the further ablation studies in Sec. 2 and comparison results on Waymo [8] and KITTI-360 [5] datasets in Sec. 4. Subsequently, Sec. 3 showcases more applications of our method.

1. Implementation details

1.1. Evaluation datasets.

We evaluate our method on the Waymo Open Dataset [8] and the KITTI-360 dataset [5]. Following previous works [9, 10, 14], we select the static and dynamic sequences from the both datasets. The specific selected sequence names and corresponding ids are listed in Tab. 1 and Tab. 2.

Table 1. **The selected sequences from the KITTI-360 [5] dataset for evaluation.** $\mathcal{S}$ and $\mathcal{D}$ denote static and dynamic sequences, respectively.

Scene name	Type	Sequence id	Start frame	End frame
ks1	$\mathcal{S}$	Seq 1538-1601	1538	1601
ks2	$\mathcal{S}$	Seq 1728-1791	1728	1791
ks3	$\mathcal{S}$	Seq 1908-1971	1908	1971
ks4	$\mathcal{S}$	Seq 3353-3415	3353	3415
kd1	$\mathcal{D}$	Seq 2351-2400	2351	2400
kd2	$\mathcal{D}$	Seq 4951-5000	4951	5000
kd3	$\mathcal{D}$	Seq 8121-8170	8121	8170
kd4	$\mathcal{D}$	Seq 10201-10250	10201	10250
kd5	$\mathcal{D}$	Seq 10751-10800	10751	10800
kd6	$\mathcal{D}$	Seq 11401-11450	11401	11450

Table 2. **The selected sequences from the Waymo Open [8] dataset for evaluation.** $\mathcal{S}$ and $\mathcal{D}$ denote static and dynamic sequences, respectively.

Scene name	Type	Sequence id	Start frame	End frame
ws1	$\mathcal{S}$	Seg 113792	1	50
ws2	$\mathcal{S}$	Seg 106762	1	50
ws3	$\mathcal{S}$	Seg 177619	1	50
ws4	$\mathcal{S}$	Seg 117240	1	50
wd1	$\mathcal{D}$	Seg 108305	148	197
wd2	$\mathcal{D}$	Seg 132712	51	100
wd3	$\mathcal{D}$	Seg 100721	1	50
wd4	$\mathcal{D}$	Seg 105003	148	197

1.2. Baseline methods

LiDARsim and PCGen. LiDARsim [6] and PCGen [4] are surfel-based reconstruction methods. Since the official implementation is not publicly available, we re-implement these two methods based on the codebase provided by the LiDAR-NeRF [9] and follow the same experimental settings on the KITTI-360 dataset [5].

LiDAR-NeRF. LiDAR-NeRF [9] is the first NeRF-based method for LiDAR re-simulation, we directly adopt the official implementation. For KITTI-360 dynamic sequences and Waymo [8] scenes, we adjust the scene scales and LiDAR resolutions for fair comparison.

LiDAR4D. LiDAR4D [14] utilizes a 4D hybrid representation combined with multi-planar and grid features for LiDAR re-simulation. We adopt the official implementation and follow the same experimental settings as their paper on the KITTI-360 dataset. For the Waymo dataset, we preprocess the dataset following the same procedure as LiDAR4D and adjust the LiDAR resolutions. The ray-drop refinement is also conducted for evaluation sequences.

DyNFL. DyNFL [10] leverages the bounding boxes of moving objects to construct an editable neural field for high-fidelity re-simulation of LiDAR scans. We follow the original implementation based on NFL Studio [2] and the settings for the Waymo dataset.

1.3. Gaussian Densification.

We adopt the adaptive control techniques from 3DGS [3] during optimization, which includes operations such as pruning, cloning, and splitting. However, unlike the vanilla 3DGS [3], which tracks screen-space gradients of particle positions for cloning and splitting decisions, our approach utilizes gradients in 3D world-space. This method is more general and suitable in our ray tracing context since the forward and backward passes are performed in 3D space. Furthermore, to prevent object Gaussians from expanding into occluded areas, we follow the strategy of [11] and sample a set of points for each object model to form a probability distribution function. During optimization,

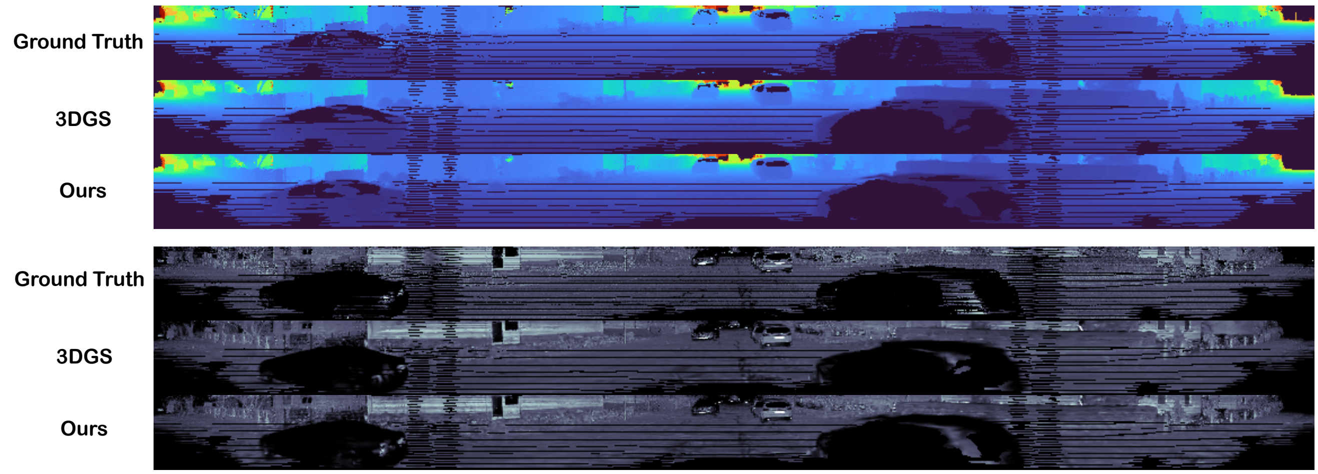

Figure 2. **Qualitative results of ablation study on ray tracing with Gaussian variants**.

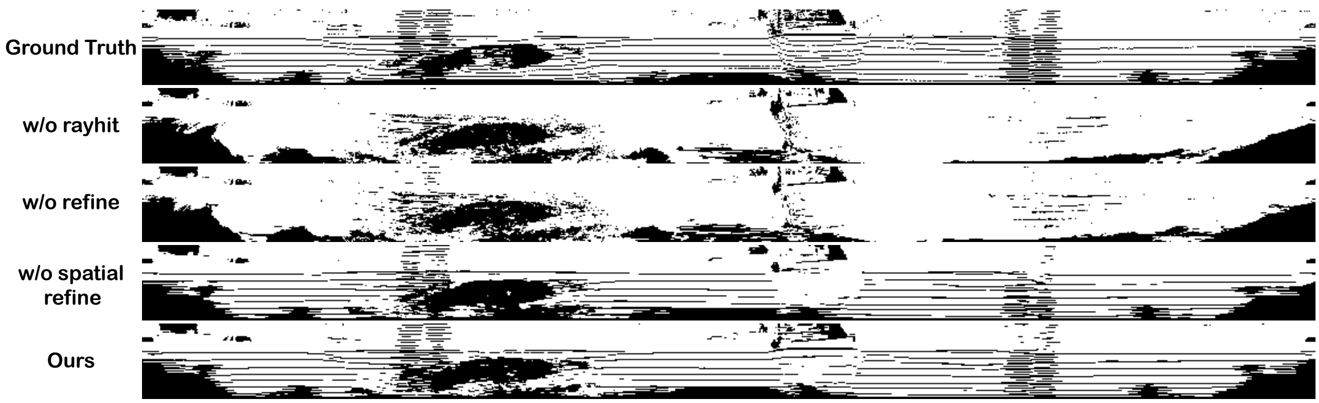

Figure 3. **Qualitative results of ablation study on ray-drop modeling and refinement**.

Gaussians associated with sampled points that fall outside the bounding box are pruned to avoid excessive growth.

1.4. Calculation of the intersection and depth.

Given the rotation matrix $\mathbf{R} = [\boldsymbol{t_u}, \boldsymbol{t_v}, \boldsymbol{t_w}]$ and mean $\boldsymbol{\mu}$ of a 2D Gaussian, the normal is calculated as $\boldsymbol{n} = \boldsymbol{t_u} \times \boldsymbol{t_v}$, then the intersection distance t is determined from the equation:

$$(\boldsymbol{r_o} + t\boldsymbol{r_d} - \boldsymbol{\mu}) \cdot \boldsymbol{n} = 0, \tag{1}$$

The pixel depth is calculated via alpha blending (Eq. 2 and Eq. 3 in main paper):

$$D = \sum_{i=1}^{K} T_i \alpha_i t_i, \tag{2}$$

where t_i denotes the i^{th} intersection distance, K is the total number of intersections.

Table 3. **Quantitative results of ablation study on ray tracing with Gaussian variants**. The cell colors present the best and the second best results, respectively.

Method	FPS↑	RMSE↓	LPIPS↓	PSNR↑	CD↓	F-score↑
3D Gaussians	29	3.6716	0.1145	27.0976	0.3553	0.8899
2D Gaussians (Ours)	42	3.4671	0.1070	27.6755	0.1077	0.9255

Table 4. **Quantitative results of ablation study on ray-drop modeling and refinement**. The cell colors present the best and the second best results, respectively.

Method	RMSE↓	LPIPS↓	PSNR↑	CD↓	F-score↑
w/o $\mathcal{R}_{\text{hit}}$	4.5482	0.4503	25.2371	0.1592	0.9089
w/o $\mathcal{R}_{\text{refine}}$	4.4635	0.4338	25.3924	0.1485	0.9119
w/o $\mathcal{R}_{\text{spatial}}$	3.7571	0.1480	26.9385	0.1247	0.9249
Ours	3.4671	0.1070	27.6755	0.1077	0.9255

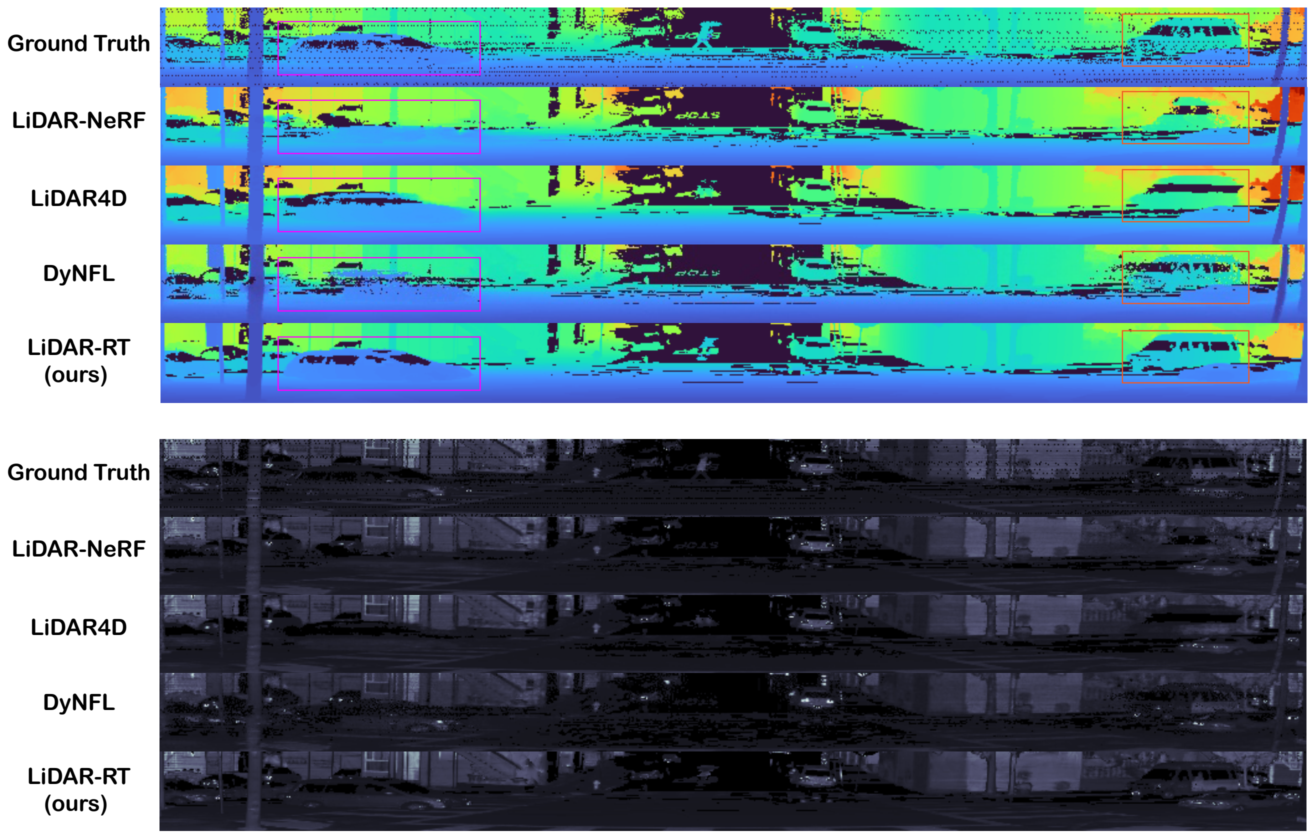

Figure 4. **Qualitative comparison of LiDAR range images on Waymo Open Dataset [8] sequence seg-108305.**

2. Ablation studies

2.1. Impact of ray tracing with Gaussian variants

Tab. 3 and Fig. 2 show the quantitative and qualitative results of the ablation study on ray tracing with Gaussian variants. We adopt 3D Gaussians [3] and 2D Gaussians [1] as our Gaussian primitives for ray tracing separately. As for 3D Gaussians, we construct the corresponding proxy geometry as an icosahedron, proposed by 3DGRT [7]. The results demonstrate that the 2D Gaussians have a slight advantage over 3D Gaussians in terms of rendering quality and efficiency, which means our ray tracer is compatible with various types of Gaussian primitives and other extensions [12, 13] applied on Gaussian primitives can be easily integrated into our framework.

2.2. Impact of ray-drop modeling and refinement

Tab. 4 and Fig. 3 present the quantitative and qualitative results of our detailed ablation study on ray-drop modeling and refinement. The variant labeled *w/o* $\mathcal{R}_{\text{hit}}$ models the ray-drop using only a single logit, results in a significant degradation of rendering quality. The variant *w/o* $\mathcal{R}_{\text{ref}}$ omits the refinement stage,consequently failing to capture the *sensor-level* ray-drop patterns. Lastly, the *w/o* $\mathcal{R}_{\text{spatial}}$ variant disregards the ray information $(\mathbf{r}_o, \mathbf{r}_d)$ as UNet inputs, leading to a loss of details on dynamic objects.

Table 5. **Evaluation results of dynamic actors**. The subscripts *int* and *depth* denote the intensity and depth, respectively.

Method	PSNR$_{int}$	SSIM$_{int}$	PSNR$_{depth}$	SSIM$_{depth}$
DyNFL [48]	30.96	0.9608	23.35	0.9552
Ours	32.51	0.9711	22.98	0.9351

3. Applications

Object decomposition. Fig. 6 illustrates the object decomposition results on Waymo dataset [8]. Our method is capable of decomposing the foreground dynamic objects clearly and produces high fidelity rendering results. We evaluate the dynamic actors separately, the quantitative results are shown in Tab. 5. Since our method based on the explicit representation of dynamic objects, it achieves better performance in terms of quality compared to the implicit methods [10].

Semantic Segmentation. We show the semantic segmentation results on the Waymo dataset [8] in Fig. 7. Similar

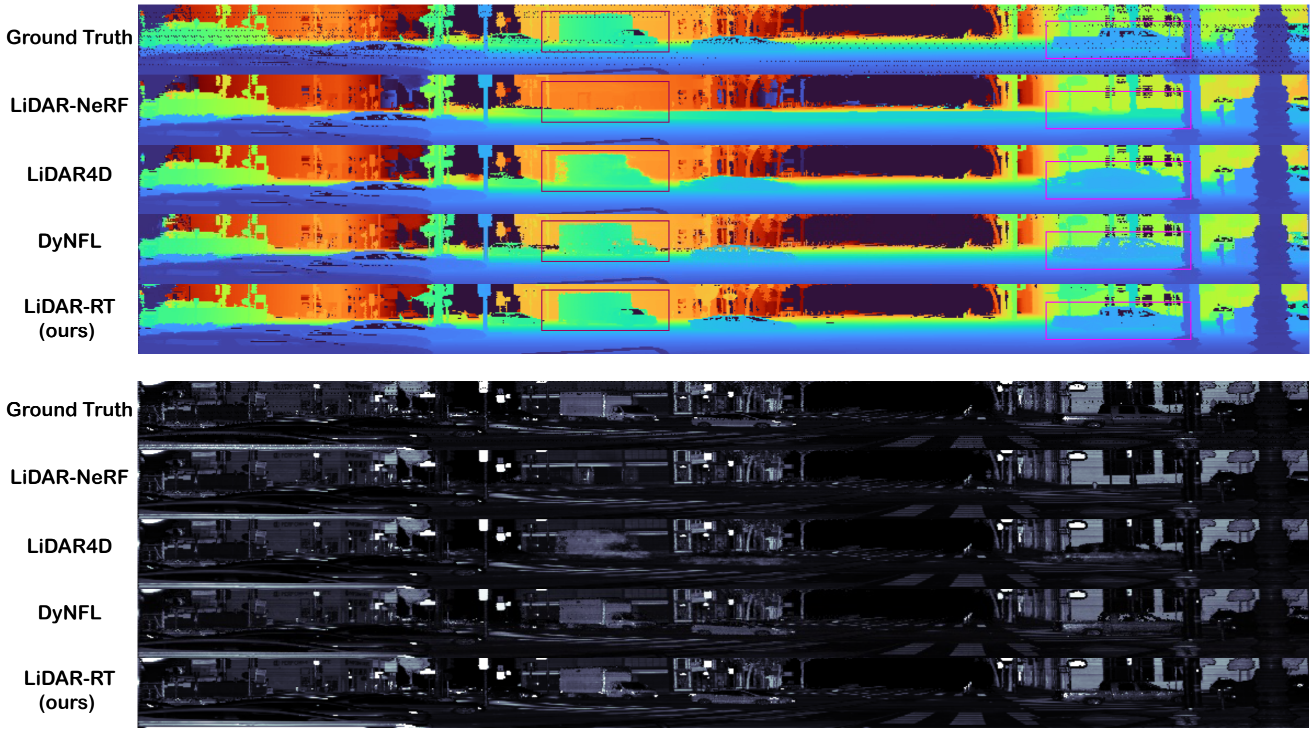

Figure 5. **Qualitative comparison of LiDAR range images on Waymo Open Dataset [8] sequence seg-132712.**

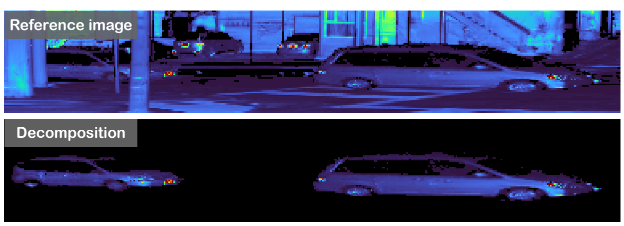

Figure 6. **Decomposition results on Waymo dataset [8].** The points are colorized by intensity values from blue(0) to red (1).

to Street Gaussians [11], our method can be easily extended to render semantic maps by assigning additional semantic attributes to the Gaussian primitives.

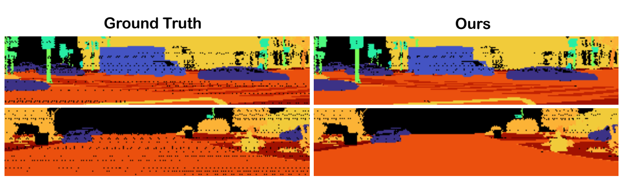

Figure 7. **Semantic segmentation results on Waymo dataset [8].** Our method is easily extended to render other feature maps such as semantics.

4. Additional results

Novel LiDAR view synthesis on the Waymo dataset. We provide additional qualitative results on the Waymo dataset [8] with multiple baselines, as shown in Fig. 1, Fig. 5, and Fig. 4. The dynamic vehicles are highlighted with colored bounding boxes (□/□) for better visualization. Even on the challenging Waymo [8] dataset with multiple moving actors and the complex urban environment, our LiDAR-RT still generates realistic novel LiDAR views with accurate geometry and high-frequency details of dynamic objects. In contrast, LiDAR-NeRF [9] struggles with dynamic objects due to its lack of temporal modeling. LiDAR4D [14] produces blurry and distorted results on this challenging dataset. While DyNFL [10] renders plausible results, also exhibits some artifacts around the dynamic objects due to the inaccurate estimations of ray-drop.

Depth comparison with rasterization. To compare the geometry accuracy of our ray tracing pipeline with rasterization, we project the 3D point clouds to the camera space and evaluate the depth map. The quantitative results are shown in Tab. 6. Our ray tracing pipeline have better geometric accuracy than rasterization.

Table 6. **Depth comparison results with rasterization**.

Method	RMSE ↓	MedAE ↓	PSNR ↑
Rasterization	2.5304	0.4091	30.9522
Ours	2.3188	0.3761	31.1296

References

[1] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 3

[2] Shengyu Huang, Zan Gojcic, Zian Wang, Francis Williams, Yoni Kasten, Sanja Fidler, Konrad Schindler, and Or Litany. Neural LiDAR fields for novel view synthesis. In *ICCV*, 2023. 1

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *TOG*, 42(4):139:1–139:14, 2023. 1, 3

[4] Chenqi Li, Yuan Ren, and Bingbing Liu. PCGen: Point cloud generator for lidar simulation. In *ICRA*, pages 11676–11682, 2023. 1

[5] 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. 1

[6] 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 *CVPR*, pages 11167–11176, 2020. 1

[7] Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3D Gaussian Ray Tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics and SIGGRAPH Asia*, 2024. 3

[8] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2446–2454, 2020. 1, 3, 4

[9] Tang Tao, Longfei Gao, Guangrun Wang, Yixing Lao, Peng Chen, Hengshuang Zhao, Dayang Hao, Xiaodan Liang, Mathieu Salzmann, and Kaicheng Yu. LiDAR-NeRF: Novel lidar view synthesis via neural radiance fields. *arXiv preprint arXiv:2304.10406*, 2023. 1, 4

[10] Hanfeng Wu, Xingxing Zuo, Stefan Leutenegger, Or Litany, Konrad Schindler, and Shengyu Huang. Dynamic lidar re-simulation using compositional neural fields. In *CVPR*, 2024. 1, 3, 4

[11] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians for modeling dynamic urban scenes. In *ECCV*, 2024. 1, 4

[12] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. AbsGS: Recovering fine details for 3d gaussian splatting, 2024. 3

[13] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3d gaussian splatting. *CVPR*, 2024. 3

[14] Zehan Zheng, Fan Lu, Weiyi Xue, Guang Chen, and Changjun Jiang. LiDAR4D: Dynamic neural fields for novel space-time view lidar synthesis. In *CVPR*, pages 5145–5154, 2024. 1, 4