LiDAR4D: Dynamic Neural Fields for Novel Space-time View LiDAR Synthesis

Supplementary Material

In this document, we start with the **ablation study** in Appendix A to demonstrate the effectiveness of the proposed key modules as a complement. Following this, we conduct a more comprehensive **analysis of the qualitative and quantitative experiments** based on Section 4 and provide additional experimental results in Appendix B. Specific **implementation details** and dataset information are subsequently presented in Appendix C for reproduction. Finally, we showcase further **applications** of LiDAR4D in Appendix D, thereby highlighting its versatility, flexibility, and great potential.

A. Ablation Study

The advantages of our method in comparison to LiDAR-NeRF [39] are illustrated in Figures 4 to 6, which corresponds to the key modules of LiDAR4D, *i.e.*, dynamic reconstruction, hybrid representation, and ray-drop refinement. In order to provide a more rigorous demonstration of the efficacy of our method, we perform the ablation study for each module and present quantitative results in Table S4.

w/o CD loss w/ CD loss GT

Figure S9. **Qualitative comparison for the geometric regularization of CD loss.**

The results in the first row represent the basic version of LiDAR4D with only hash grid representation, which is similar to LiDAR-NeRF. The introduction of the hybrid representation ($\mathcal{H}.$) significantly enhances the reconstruction quality, especially for the point cloud and depth metrics in *Row 2*. Subsequently, we further adopted time-conditioned dynamic-part representations ($\mathcal{D}_{\mathcal{T}}.$) and flow-constrained temporal feature aggregation ($\mathcal{D}_{\mathcal{F}}.$), which notably strengthened the capability of dynamic reconstruction in *Row 3&4*. Among them, the incorporation of CD loss as geometric regularization benefits the optimization of flow MLP and leads to more accurate results for dynamic objects, as shown in Figure S9. Ultimately, the global optimization of ray-drop ($\mathcal{R}.$) based on U-Net assists LiDAR4D in achieving SOTA performance in the last row.

B. Additional Analysis and Experiments

B.1. Quantitative and Qualitative Comparison

Static Scenes. As shown in Figure S12, traditional explicit reconstruction methods such as LiDARsim [25] convert point cloud scenes into mesh representations but struggle to accurately reconstruct object details in large-scale scenes (*Row 2*). We additionally adopt the state-of-the-art surface reconstruction algorithm NKSR [15] upon LiDARsim to improve the reconstruction quality. Nevertheless, the novel-view results are still significantly different from the ground truth (*Row 3*). Furthermore, it is unable to establish the correlation between intensity and viewpoint. PC-Gen [19] reconstructs directly based on the point cloud, while the generated results are heavily affected by noise (*Row 4*). On the contrary, the implicit reconstruction method like LiDAR-NeRF [39] (*Row 5*) alleviates the challenges above and achieves a substantial lead. Our LiDAR4D further surpasses the previous approaches, especially in reconstruction details such as vehicle shape and window reflections (*Row 6*). The quantitative results illustrated in Table 3 demonstrate a similar trend. Compared to LiDAR-NeRF, the hybrid representation and ray-drop refinement of LiDAR4D lead to a 12.0% and 13.7% drop in the depth and intensity RMSE metrics.

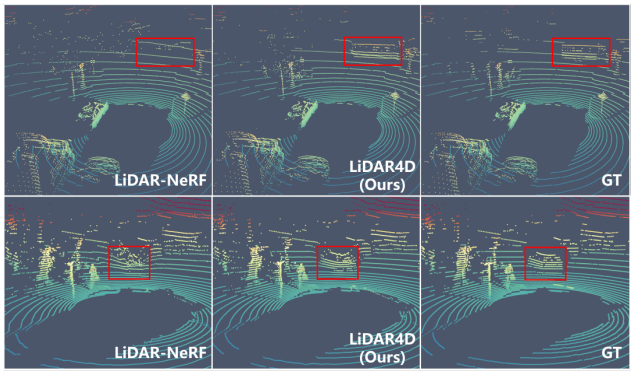

Figure S10. **Qualitative novel view LiDAR point cloud synthesis results on NuScenes dataset.**

Dynamic Scenes. Explicit reconstruction methods fail completely in dynamic scenes (Figures 7, 8, S13 and S14), which yields extremely poor validation results (Tables 1 and 2) due to the stacking of dynamic objects. In contrast, implicit reconstruction methods largely avoid the artifacts and noise of dynamic objects. However, existing methods like LiDAR-NeRF are designed for static scenes, resulting in the obscureness or absence of moving objects (Figures 6 and S10). Although D-NeRF [32] incorporates a deformation field, its impact is quite limited. The primary issue lies in the lack of constraints and the difficulty of establishing long-distance correspondence. Moreover, the state-of-the-art dynamic methods TiNeuVox [9] and K-planes [12] are limited by their representation resolution, which makes it difficult to reconstruct details in large-scale scenes, such as

$\mathcal{H}$.	$\mathcal{D}_\mathcal{T}$.	$\mathcal{D}_\mathcal{F}$.	$\mathcal{R}$.	Point Cloud		Depth					Intensity				
				CD↓	F-score↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑
✗	✗	✗	✗	0.1840	0.8979	4.0602	0.0639	0.2692	0.6483	26.1957	0.1398	0.0431	0.2969	0.3829	17.2018
✓	✗	✗	✗	0.1429	0.9116	3.9702	0.0499	0.2586	0.6645	26.3647	0.1368	0.0411	0.2760	0.4036	17.3675
✓	✓	✗	✗	0.1213	0.9221	3.6947	0.0448	0.2397	0.7027	27.0285	0.1286	0.0368	0.2688	0.4553	17.8999
✓	✓	✓	✗	0.1187	0.9260	3.6745	0.0425	0.2130	0.7104	27.1009	0.1281	0.0359	0.2426	0.4726	17.9394
✓	✓	✓	✓	**0.1089**	**0.9272**	**3.5256**	**0.0404**	**0.1051**	**0.7647**	**27.4767**	**0.1195**	**0.0327**	**0.1845**	**0.5304**	**18.5561**

Table S4. **Ablation study on KITTI-360 Dataset**. $\mathcal{H}$: hybrid representation, $\mathcal{D}_\mathcal{T}$: time-conditioned dynamic-part representations, $\mathcal{D}_\mathcal{F}$: flow-constrained temporal feature aggregation, $\mathcal{R}$: global ray-drop refinement.

w/ GT Mask		Depth			Intensity		
		LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
$\mathcal{S}$.	LiDAR-NeRF	0.025	0.971	34.808	0.146	0.667	19.935
	Ours	**0.024**	**0.974**	**36.303**	**0.145**	**0.704**	**20.677**
$\mathcal{D}$.	LiDAR-NeRF	0.126	0.843	29.361	0.192	0.583	18.891
	Ours	**0.019**	**0.981**	**36.222**	**0.137**	**0.715**	**21.407**

Table S5. **Experiments with GT ray-drop mask on KITTI-360 Dataset**. $\mathcal{S}$: Static sequences, $\mathcal{D}$: Dynamic sequences.

vehicle and pedestrian geometry (*Row 7&8* in Figure 8), as well as high-frequency details in intensity (*Row 7&8* in Figure S13). Our proposed LiDAR4D instead accomplishes geometric-aware and time-consistent dynamic reconstruction through 4D hybrid representation and flow-constrained temporal feature aggregation. As shown in Tables 1 and 2, LiDAR4D ranks first across almost all metrics. A considerable visualization intuitively exhibits the superior generation quality of LiDAR4D, encompassing both long-distance moving vehicles and small bicyclists (the last row in Figures 8 and S14).

Difference on Ray-drop. Existing methods differ in ray-drop modeling. PCGen [19] employs MLP to estimate ray-drop, while LiDARsim [25] adopts U-Net, which takes depth and intensity values as input. In contrast, LiDAR4D predicts the ray drop probability of each point in space through neural fields and integrates them along the ray as the inputs of U-Net. Then, the U-Net is optimized in runtime to refine the prediction for individual scenarios. As can be seen from Figure S12, the MLP-based method may handle high-frequency details, but it also results in noisy prediction (*Row 4&5*). The U-Net-based method preserves global patterns better (*Row 2&3*) and consequently achieves superior results in LPIPS [45] metrics in Tables 1 and 3 in particular. However, this data-driven paradigm is dependent on the distribution of the training samples and is difficult to predict accurately in detail, *i.e.*, the vehicle windows. LiDAR4D combines the advantages of both to achieve more realistic ray-drop modeling, as shown in Figure 5.

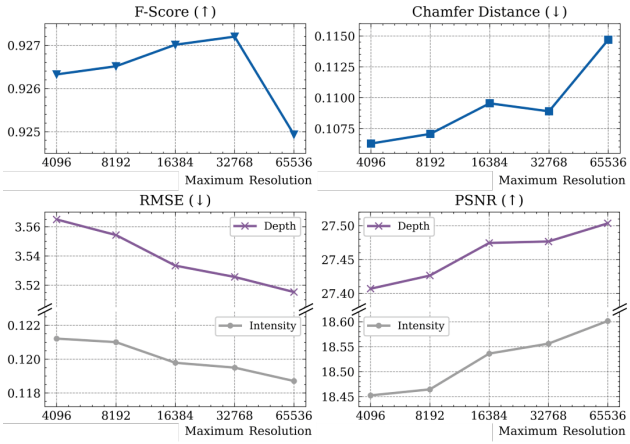

Figure S11. **Influence of maximum representation resolution.**

B.2. Experiments without Ray-drop Effect

In order to eliminate the effect of ray-drop on the evaluation metrics, we conduct supplementary experiments by only calculating the results on rays that have valid values. In other words, we apply the ground-truth ray-drop mask to all results for reconstruction quality evaluation. As shown in Table S5, our method outperforms LiDAR-NeRF [39] in both static and dynamic scenarios, especially by a large margin in dynamic sequences.

B.3. Experiments on Resolution

Increasing the resolution of the representations is important for large-scale scenarios. In comparison to dense grids and planar features, hash grids can substantially raise the resolution and thus improve the accuracy of reconstruction, which has been verified in previous experiments. To determine the maximum resolution of hash grids, we further conducted additional experiments. As illustrated in Figure S11, increasing the resolution continuously alleviates the error of depth and intensity reconstruction. Considering the limited capacity, an extremely high resolution may lead to unfavorable effects, such as the degradation of point cloud metrics. Finally, we select the resolution of 2^{15}, which can adequately meet the requirements of large-scale scene reconstruction.

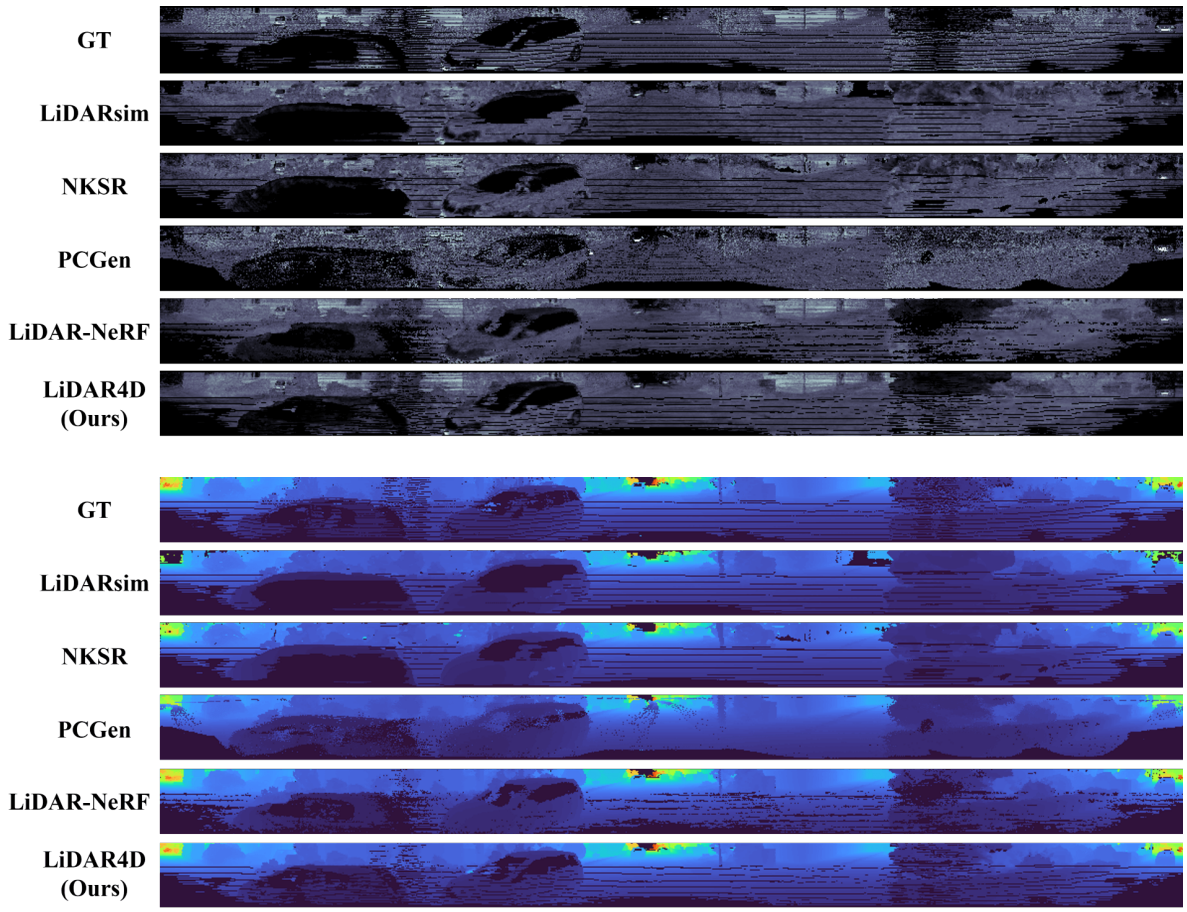

Figure S12. **Qualitative comparison on KITTI-360 *Static* Scene Sequences.**

KITTI-360		NuScenes	
Static	Seq 1538-1601		Seq 450-500
	Seq 1728-1791		
	Seq 1908-1971		Seq 1250-1300
	Seq 3353-3416		(ego-vehicle stationary)
Dynamic	Seq 2350-2400	**Dynamic**	Seq 1600-1650
	Seq 4950-5000		
	Seq 8120-8170		Seq 2200-2250
	Seq 10200-10250		
	Seq 10750-10800		Seq 3180-3230
	Seq 11400-11450		

Table S6. **Scene sequences of KITTI-360 and NuScenes**.

C. Implementation Details

C.1. Dataset Visualization

As shown in Figure S17, we selected 6 representative dynamic scene sequences on KITTI-360. Each scene spans a distance of about 100–200 m and contains vehicles or pedestrians moving over long distances. Following the setup of LiDAR-NeRF [39], the same experiments were conducted on the original 4 static scene sequences (Figure S18). The height and width of the range images are 66 and 1030. For the NuScenes dataset, we chose 5 dynamic sequences illustrated in Figure S19, including an ego-vehicle stationary scene (*Column 2*) which can be viewed as a special case of novel temporal view synthesis. The size of the range images is set to 32 $\times$ 1080. The substantial variations between scenarios serve as a more accurate indicator of the reconstruction capabilities of current methods. The index number of scene sequences can be found in Table S6.

C.2. LiDAR4D

Hybrid representation. For the multi-planar features, the base resolution of the spatial plane is set to 64. The multi-scale structure has 4 levels, each doubling the spatial resolution and output 8-dimension feature, which finally yields a 32-dimension feature for both static and dynamic parts. The spatial resolution of hash grids ranges from 512 (the maximum resolution of multi-planar features) to 2^{15}. There are a total of 8 levels of hash grids, each level outputs 4-dimension features, and then the same 32-dimension fea-

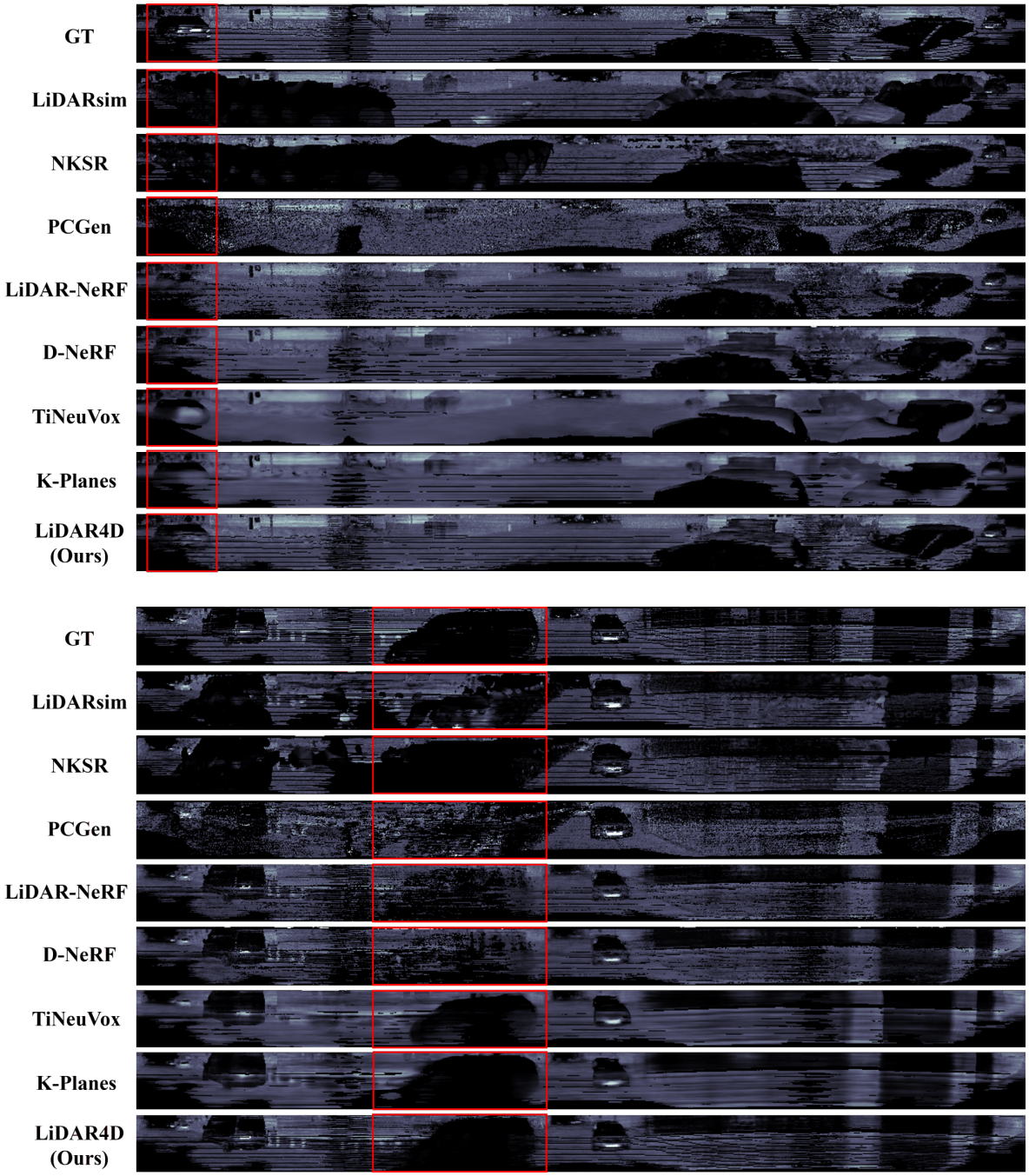

Figure S13. **Qualitative comparison for LiDAR intensity reconstruction and synthesis.** *Dynamic* vehicles are marked with red boxes.

tures are obtained. The grid is mapped to a hash table of 2^{19}. All the temporal resolution is fixed to 25. Ultimately, the static and dynamic features of the planes and hash grids compose a 128-dimensional latent vector.

Dynamic modules. Beyond time-conditioned multi-planar and hash grid features for dynamic reconstruction, we additionally introduce flow MLP to aggregate dynamic features for temporal consistency. This coordinate-based MLP is an 8-layer neural field with 128 units per layer. Eventually, the dynamic features of adjacent spatio-temportal points are aggregated by weighted averaging. We incorporate the chamfer distance loss based on point clouds to effectively constrain the optimization of the flow MLP. It encourages the two adjacent frames of the point cloud transformed by the

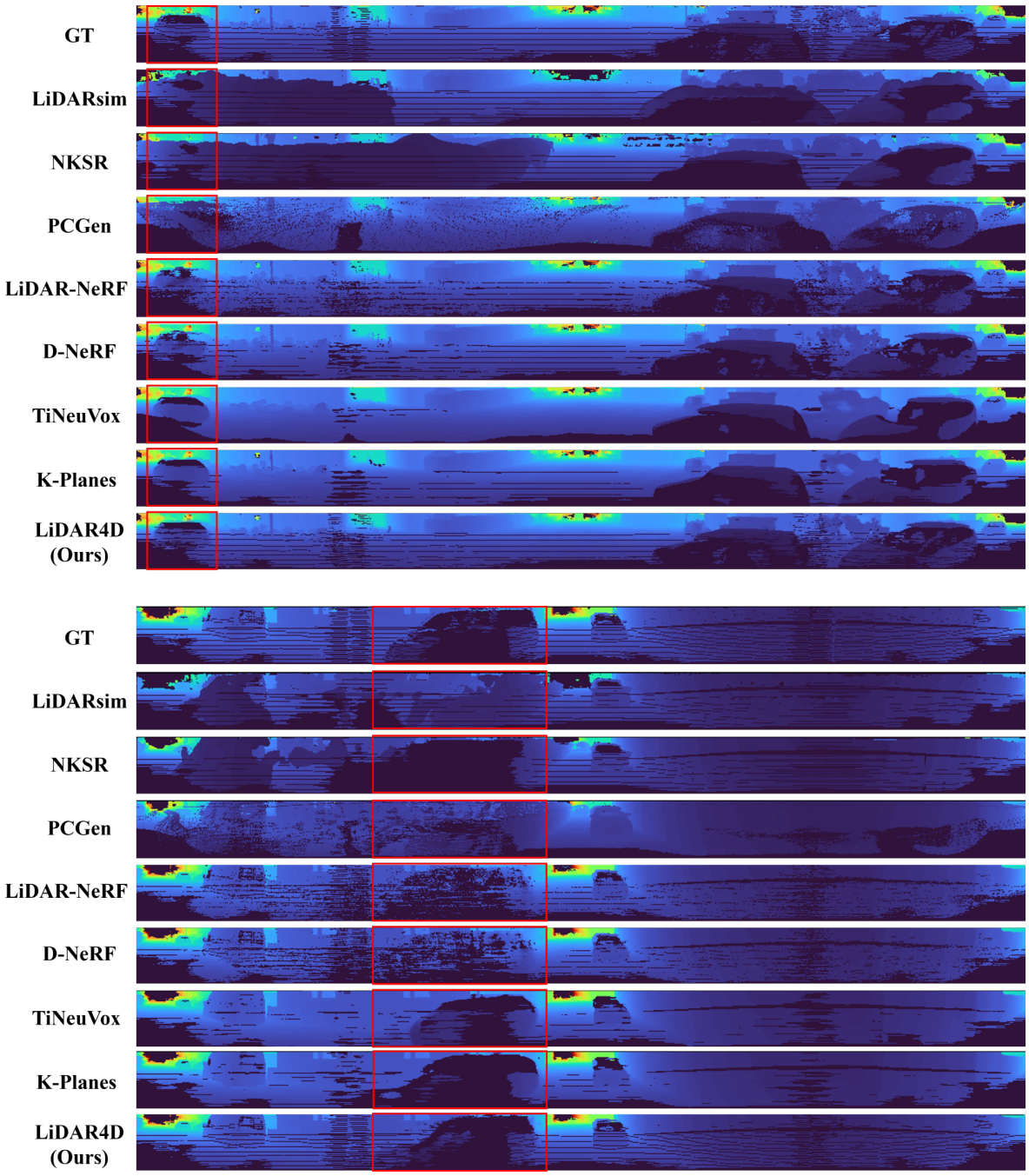

Figure S14. **Qualitative comparison for LiDAR depth reconstruction and synthesis.** *Dynamic* vehicles are marked with red boxes.

predicted scene flow to be as consistent as possible. According to the training process, we randomly select one moment in each epoch for optimization. In addition, we preprocess the point cloud by removing ground points using RANSAC [10] and further limiting the maximum distance within 50 meters to mitigate the adverse effect of noise.

Neural LiDAR fields. The aggregated time-conditioned and flow-constrained dynamic features are finally fed into a 2-layer 64-dimensional MLP, which outputs the 15-dimensional geometric feature and density value. The geometric feature with the 12-band frequency-encoded viewpoint is utilized to predict intensity values and ray-drop probabilities by two independent 3-layer 64-dimensional MLPs, respectively. The expectation of the density in-

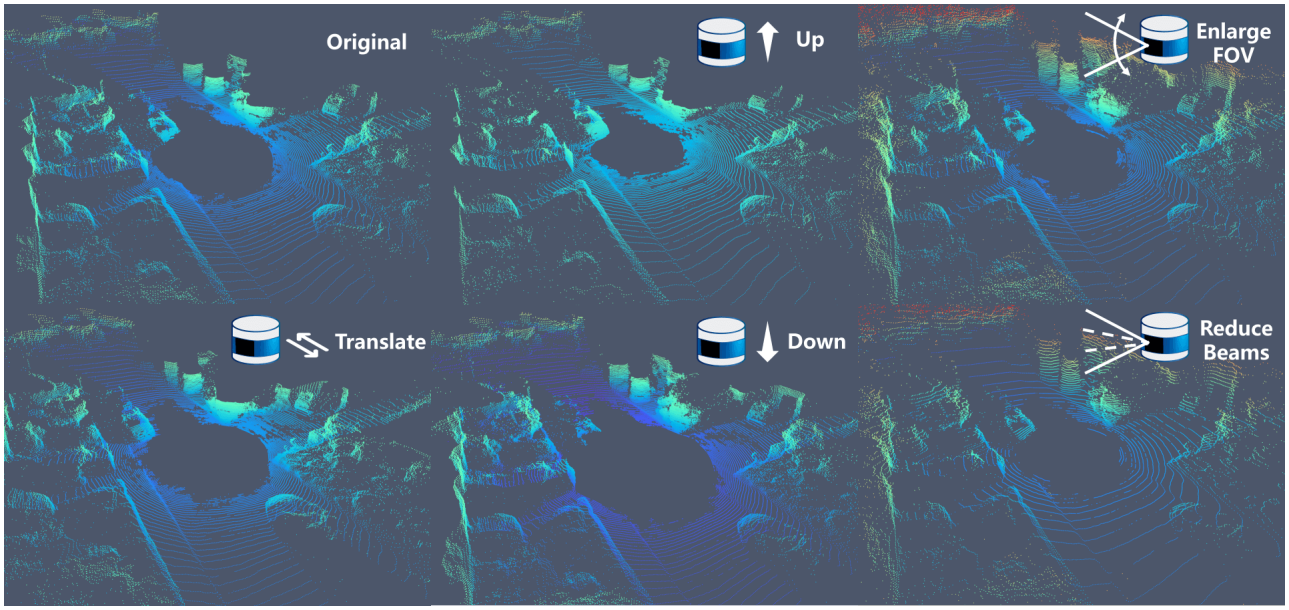

Figure S15. **Novel view point cloud synthesis with different LiDAR configurations.**

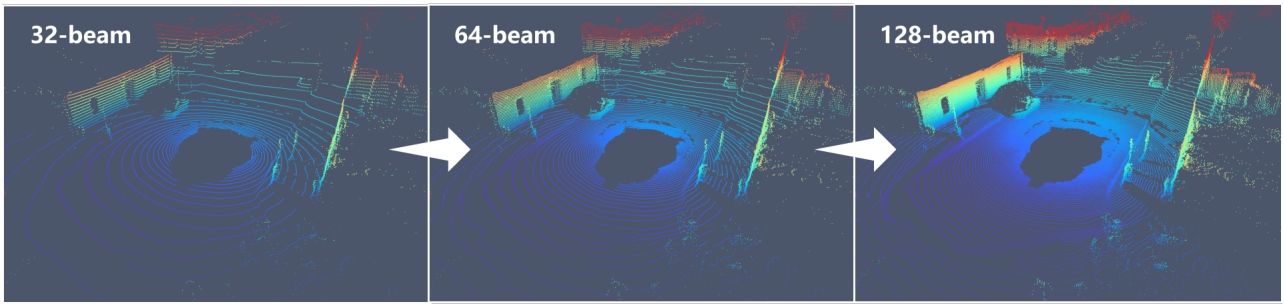

Figure S16. **Increase LiDAR beams to densify the point cloud on NuScenes dataset.**

tegrated along the ray serves as the depth value. Then, these initial predictions are combined as inputs to U-Net for global ray-drop optimization. The final predictions are multiplied by the ray-drop mask for synthesis.

Optimization details. The initial learning rate is set to 0.01 for the multi-planar and hash grids, and 0.001 for other MLP networks. The learning rate decreases exponentially during iterations, with a final decay coefficient of 0.1. The depth-loss weight λ_α is 1, the intensity weight λ_β is 0.1, the ray-drop weight λ_γ is 0.01, and the flow weight λ_η is 0.01. During refinement, the weight λ_r is set to 1 with other loss weights set to 0. The U-Net weights are randomly initialized and optimized with a learning rate of 0.001 by the Adam [17] optimizer. Other unmentioned optimization details are basically in line with LiDAR-NeRF [39].

Efficiency. According to experiments conducted on a single NVIDIA GeForce RTX 4090 GPU, LiDAR4D takes about 2 hours to complete the optimization of each scenario.

D. Applications

At last, we showcase the application of LiDAR4D for novel-view LiDAR synthesis with different sensor configurations. As illustrated in Figure S15, we can freely manipulate the sensor's pose, *e.g.*, moving up and down or horizontal translation. It can be observed that the LiDAR point clouds under different sensor poses vary significantly, and the accurate recovery of the scene and objects further demonstrates the high-fidelity synthesis of LiDAR4D. In addition, we can adjust LiDAR configurations, such as increasing the vertical field of view, to obtain a wider range of sensing results on the top right of Figure S15. Alternately, the modification of LiDAR beams realizes the conversion between sparse and dense point clouds. As shown on the bottom right of Figure S15, we transfer the LiDAR configuration of KITTI-360 to that of NuScenes, realizing the crossing of the domain gap. Also as shown in Figure S16, we can also densify the sparse Nuscenes data, which will also be beneficial for downstream tasks. All of this reveals the adaptability and enormous potential of LiDAR4D.

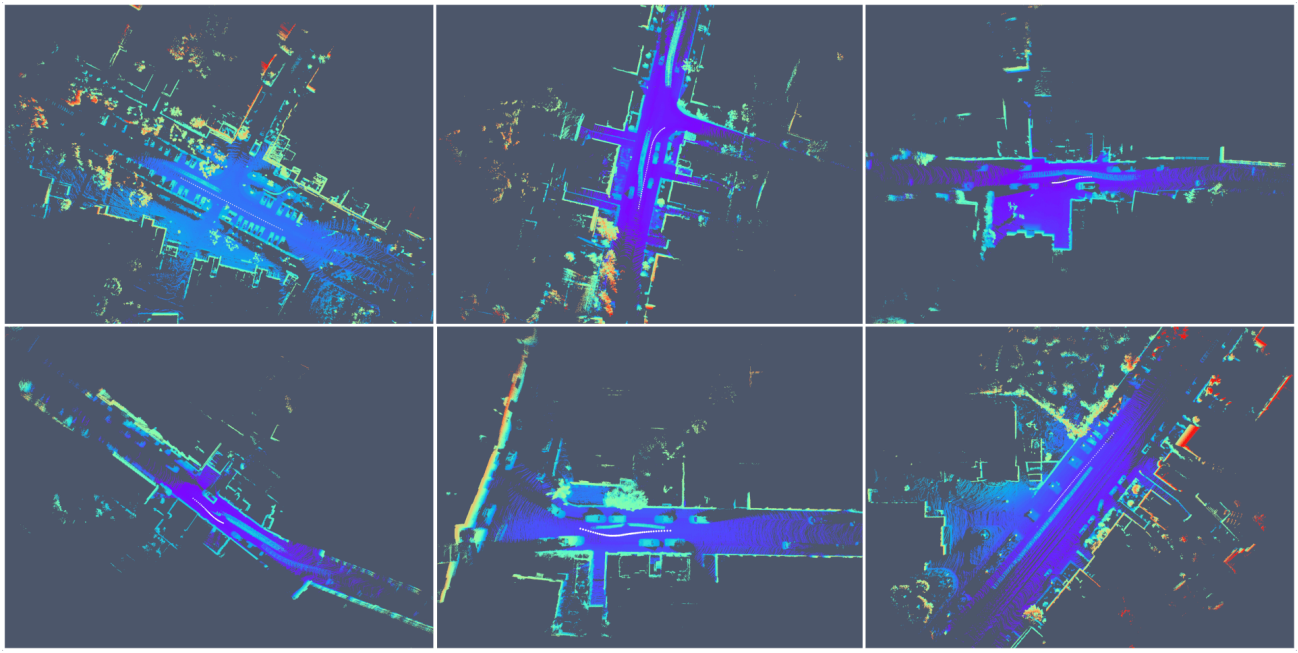

Figure S17. **Visualization for the *dynamic* sequences of KITTI-360 dataset.**

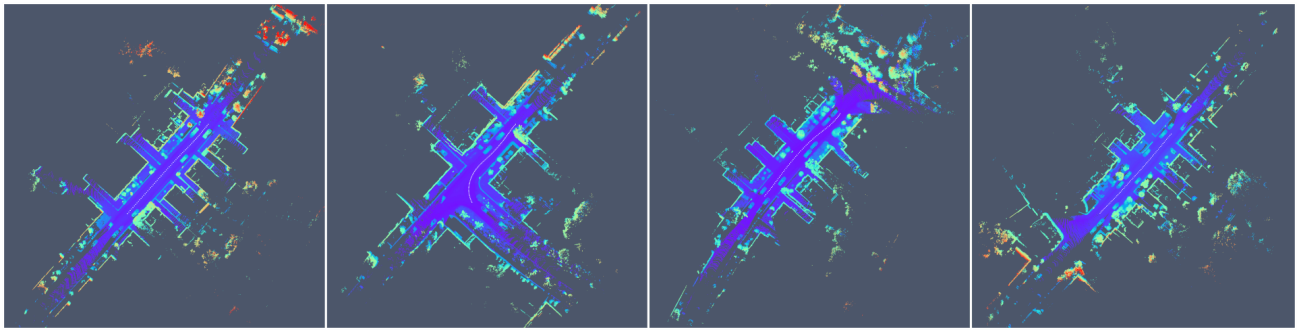

Figure S18. **Visualization for the *static* sequences of KITTI-360 dataset.**

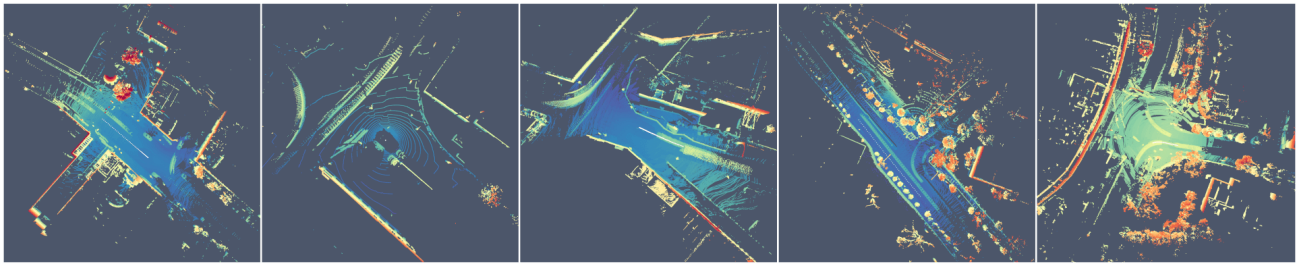

Figure S19. **Visualization for the *dynamic* sequences of NuScenes dataset.**