STGC-NeRF: Spatial-Temporal Geometric Consistency for LiDAR Neural Radiance Fields in Dynamic Scenes

Shangshu Yu[*1], **Xiaotian Sun**[*2,3], **Wen Li**[*2,3], **Qingshan Xu**[1†], **Zhimin Yuan**[2,3], **Sijie Wang**[1], **Rui She**[4], **Cheng Wang**[2,3†]

[1]Nanyang Technological University, Singapore
[2]Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, China
[3]Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, China
[4]Beihang University, China
shangshu.yu@ntu.edu.sg

Abstract

While Neural Radiance Fields (NeRFs) have advanced the frontiers of novel view synthesis (NVS) using LiDAR data, they still struggle in dynamic scenes. Due to the low frequency and sparsity characteristics of LiDAR point clouds, it is challenging to spontaneously learn a dynamic and consistent scene representation from posed scans. In this paper, we propose STGC-NeRF, a novel LiDAR NeRF method that combines spatial-temporal geometry consistency to enhance the reconstruction of dynamic scenes. First, we propose a temporal geometry consistency regularization to enhance the regression of time-varying scene geometries from low-frequency LiDAR sequences. By estimating the pointwise correspondences between synthetic (or real) and real frames at different times, we convert them into various forms of temporal supervision. This alleviates the inconsistency caused by moving objects in dynamic scenes. Second, to improve the reconstruction of sparse LiDAR data, we propose spatial geometric consistency constraints. By computing multiple neighborhood feature descriptors incorporating geometric and contextual information, we capture structural geometry information from sparse LiDAR data. This helps encourage consistent direction, smoothness, and detail of the local surface. Extensive experiments on the KITTI-360 and nuScenes datasets demonstrate that STGC-NeRF outperforms state-of-the-art methods in both geometry and intensity accuracy for dynamic LiDAR scene reconstruction.

Code — https://github.com/PSYZ1234/STGC-NeRF

Introduction

Novel view synthesis (NVS) for LiDAR data in dynamic scenes involves learning time-varying scene representations from a set of posed LiDAR scans to generate unseen frames. It is crucial for various applications in autonomous driving (Li et al. 2024; Yu et al. 2024; Fang et al. 2024a) and robotics (Dong et al. 2019; Fang et al. 2023, 2024b). Conventional LiDAR simulators (Manivasagam et al. 2020; Yang et al. 2023) usually overlook the physical properties of LiDAR sensors, leading to geometric errors in NVS.

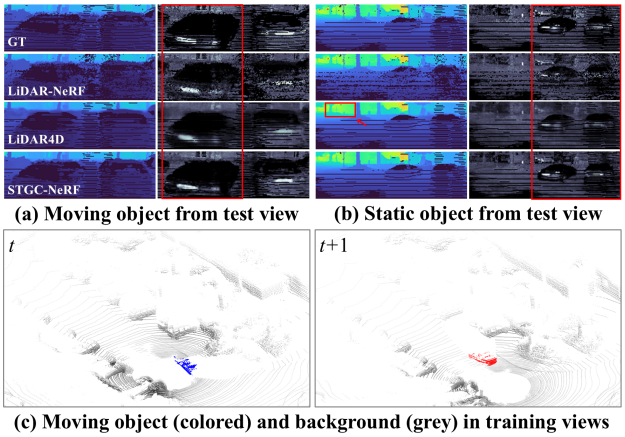


Figure 1: Outdoor rendering results (a) (b) for LiDAR depth (left) and intensity (right). Compared to our STGC-NeRF, LiDAR-NeRF (Tao et al. 2023) and LiDAR4D (Zheng et al. 2024) struggle to reconstruct accurate results in dynamic regions and sparse areas (red boxes). (c) is the LiDAR point cloud from different timestamps t and $t+1$.

With the unprecedented success of neural radiance fields (NeRFs) (Mildenhall et al. 2021), it has emerged as a promising way for LiDAR NVS and 3D scene reconstruction. LiDAR NeRFs (Tao et al. 2023; Huang et al. 2023; Hu et al. 2024; Zhang et al. 2024; Sun et al. 2024a; Tao et al. 2024) implicitly represent the 3D scene and synthesize novel views through volume rendering within a continuous representation space. Compared to LiDAR simulators, LiDAR NeRFs consider two-way transmittance and beam width of LiDAR, enabling high-fidelity LiDAR scene synthesis.

Despite the initial success, LiDAR NeRFs still struggle to accurately re-simulate dynamic real-worlds (Tao et al. 2023; Huang et al. 2023). Recent LiDAR NeRFs (Zheng et al. 2024; Wu et al. 2024) propose implicit dynamic scene representation by using 4D space-time or background-object

[*]These authors contributed equally.
[†]Corresponding author

modeling, respectively. However, they still struggle with long-distance vehicle motion and sparse point reconstruction, as illustrated in Fig. 1 (a) and (b). Since LiDAR point clouds are low frequency in time and sparse in space, it greatly reduces the potential consistency between multiple training views, as shown in Fig. 1 (c). When a LiDAR sensor scans a fast moving object at a frequency of 10 to 20 Hz, there is a large displacement of the object even in two consecutive frames. Meanwhile, sensor properties mean that the point cloud becomes less dense the further from the sensor. The above reasons together limit the performance of LiDAR NeRFs on dynamic scene reconstruction.

In this paper, we propose a novel LiDAR NeRF method, STGC-NeRF, which integrates spatial-temporal geometry consistency to improve dynamic scene reconstruction. We propose two novel designs that regularize the scene representation learning in both the temporal and spatial domains. First, we propose Temporal Geometric Consistency Regularization (TGCR) to enhance dynamic scene representation. Specifically, we estimate scene flow between synthetic (or real) and real frames at different times with a pre-trained model to obtain pointwise correspondence. We then convert them into different temporal supervisions to regularize the time-varying scene geometry regression in low-frequency LiDAR frame sequences. Second, to enhance the reconstruction details of sparse LiDAR data, we propose Spatial Geometric Consistency Constraint (SGCC). We compute various neighborhood feature descriptors, i.e., fundamental geometric and contextual features, for synthetic and real frames. By constraining feature learning, we ensure that the local reconstructed surface maintains consistent direction, smoothness, and detail. We conducted extensive experiments on KITTI-360 (Liao, Xie, and Geiger 2022) and nuScenes (Caesar et al. 2020) datasets, and the results show that STGC-NeRF has great advantages over the state-of-the-art in geometry and intensity reconstruction for complex dynamic scenes.

Our contributions are summarized as follows:

- We propose STGC-NeRF, which effectively regularizes geometric consistency across two important dimensions: temporal and spatial, for dynamic LiDAR NeRFs.
- Novel temporal geometric consistency regularization, which enhances the regression of time-varying scene geometries from low-frequency LiDAR sequences.
- Innovative spatial geometric consistency constraint, which improves the local reconstruction details of sparse LiDAR data by integrating neighborhood geometric and contextual feature constraints.
- Extensive experiments on KITTI-360 and nuScenes datasets demonstrate the great effectiveness of our methods. In particular, our method outperforms state-of-the-art methods by 8.4%/9.8% on Chamfer Distance error.

Related Work

LiDAR Simulation. Conventional LiDAR simulators (Koenig and Howard 2004; Gschwandtner et al. 2011) create a 3D virtual world to render point clouds. Early works, e.g., CARLA (Dosovitskiy et al. 2017) and AirSim (Shah et al. 2018), focus on using graphics engines to perform LiDAR simulation. Nevertheless, they need costly human annotations, and a significant domain gap exists between the generated and real point clouds. Recent works (Fang et al. 2020; Yang et al. 2023) attempt to combine realistic LiDAR data to solve the above challenges. LiDARsim (Manivasagam et al. 2020) performs the reconstruction using mesh surfel (Pfister et al. 2000) representation. PCGen (Li, Ren, and Liu 2023) reconstructs the 3D scene directly from point clouds and then renders using rasterization. However, they still suffer from large noise during point cloud generation. In addition, they are only applicable to static scene simulation.

Image-based NeRFs. Recently, NeRFs (Mildenhall et al. 2021; Barron et al. 2021) have revolutionized the long-standing challenge of image-based novel view synthesis (NVS) tasks. As an implicit neural representation method, it takes multiple posed images to represent the whole 3D scene. Many variations propose to employ neural representations based on MLPs (Sun et al. 2024b), voxel grids (Liu et al. 2020), multilevel hash grids (Müller et al. 2022), vector decomposition (Chen et al. 2022), and triplanes (Hu et al. 2023) for NVS. There also emerges researches (Barron et al. 2022; Wang et al. 2023) focusing on extending object reconstruction to large-scale scene synthesis. In addition, a portion of the studies focus on reconstructing dynamic scenes. One line of research (Pumarola et al. 2021; Fang et al. 2022; Liu et al. 2023) models the dynamic scene with an additional time dimension. Another line of research (Yan, Li, and Lee 2023; Kniaz et al. 2023) uses individual MLPs to represent dynamic objects and static backgrounds.

LiDAR-based NeRFs. Although NeRFs have made great advances in the field of imagery, there is still little research in the field of LiDAR. For these methods, NeRF-LiDAR (Zhang et al. 2024), LidaRF (Sun et al. 2024a), and AlignMiF (Tao et al. 2024) are LiDAR-camera joint synthesis methods. They incorporate multimodal inputs to perform the scene reconstruction. However, multimodal data are not always available in many cases. LiDAR-NeRF (Tao et al. 2023) and NFL (Huang et al. 2023) are LiDAR-only synthesis methods. They employ neural rendering for depth, intensity, and ray-drop probability reconstruction, synthesizing novel LiDAR views. Nevertheless, these methods cannot synthesize dynamic scenes well.

For LiDAR NeRFs in dynamic scenes, LiDAR4D (Zheng et al. 2024) and DyNFL (Wu et al. 2024) are the only two studies. LiDAR4D relies solely on point clouds with poses and times, creating a 4D hybrid representation, and further refines the final rendered result using a CNN. DyNFL uses expensive object labels (3D bounding box) to decompose the scene into a static background and dynamic objects, modeling them separately and combining them during rendering. However, due to the inherent low frequency and sparsity of LiDAR data, the reconstruction performance in dynamic scenes of these methods still leaves potential for enhancement. In this paper, we integrate novel temporal and spatial geometric consistency constraints jointly into the LiDAR NeRF network. The proposed geometric consistency overcomes the intrinsic shortcomings of LiDAR, implementing a dynamic LiDAR NeRF method using only posed scans.

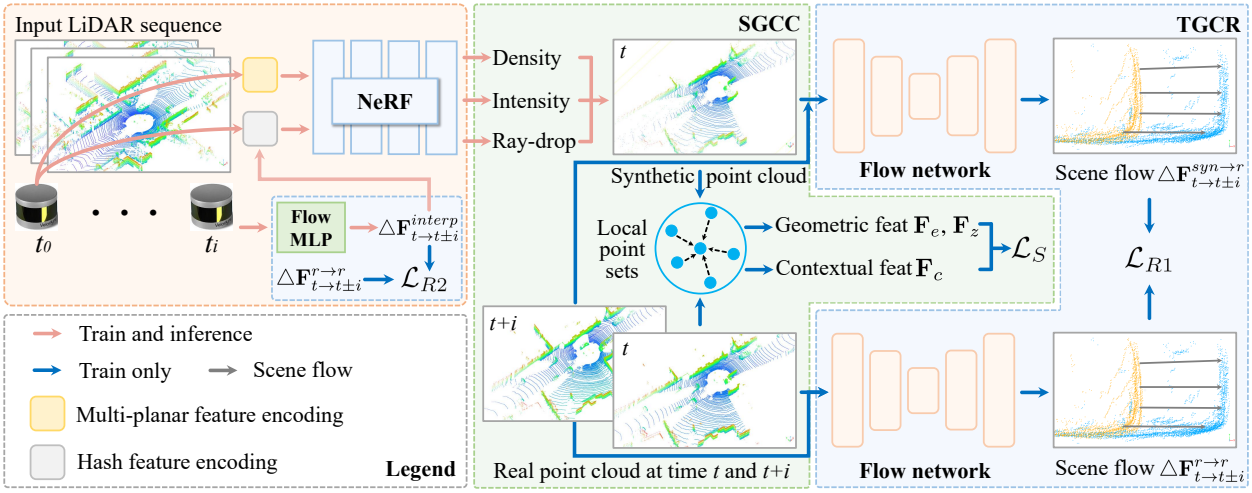


Figure 2: Overview of STGC-NeRF during training. First, the NeRF network takes the sampled 3D positions and ray directions from the LiDAR sequence as inputs, returning density, intensity, and ray-drop probability. Then, the synthetic (real) and real point clouds are fed into a pre-trained scene flow network for flow estimation. **TGCR** regularizes the time-varying scene geometry regression. Meanwhile, we extract fundamental geometric and contextual features from synthetic and real local point sets. **SGCC** constrains the neighborhood feature learning. The total network is trained in an end-to-end manner.

Methodology

Preliminaries

Problem Statement. Given LiDAR point cloud sequences $\mathcal{S} = \{S_i\}_{i=1}^{N}$ captured by moving devices, each frame of LiDAR scan $S_i \in \mathbb{R}^{K\times 4}$ is associated with a pose $P_i \in SE(3)$ and a timestamp $t_i \in \mathbb{R}$. S_i includes K 3D point coordinates and 1D reflection intensity, and N is the total frame numbers. For LiDAR novel view synthesis (NVS) in dynamic scenes, our aim is to model the scene as an implicit neural representation. Then, we can synthesize any dynamic LiDAR scans S_{syn} from an arbitrary new viewpoint P_{new}.

Neural Radiance Fields (NeRFs) Fundamentals. NeRFs take a 3D location x and a viewing direction θ as input, learning an implicit function to estimate the volume density σ and color $\mathbf{c}$. Specifically, given rays $\mathbf{r}$ originating from the sensor origin $\mathbf{o}$ in the direction $\mathbf{d}$, i.e., $\mathbf{r} = \mathbf{o} + t\mathbf{d}$, the volume rendering can be depicted as follows:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} w_i \mathbf{c}_i, \text{ with } w_i = T_i(1 - e^{(-\sigma_i \delta_i)}), \quad (1)$$

where $T_i = e^{(-\sum_{j=1}^{i-1} \sigma_j \delta_j)}$ is the accumulated transmittance, δ indicates the distance between adjacent point samples. To render LiDAR scans via NeRFs, recent works (Tao et al. 2023; Zheng et al. 2024) treat the oriented laser beams as ray sets. $\mathbf{o}$ is the LiDAR center and $\mathbf{d}$ is the normalized direction vector of the beam. The depth measurement $\hat{D}(\mathbf{r})$, point cloud intensity $\hat{I}(\mathbf{r})$, and ray-drop probability $\hat{P}(\mathbf{r})$ of LiDAR can be expressed as follows:

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^{N} w_i d_i, \hat{I}(\mathbf{r}) = \sum_{i=1}^{N} w_i i_i, \hat{P}(\mathbf{r}) = \sum_{i=1}^{N} w_i p_i, \quad (2)$$

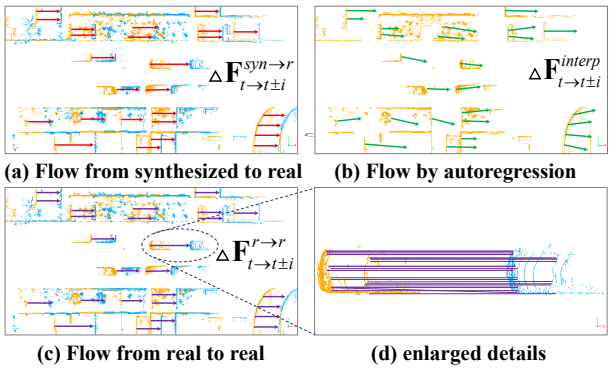
where d_i, i_i, and p_i are depth, intensity, and ray-drop probability at the i-th sample along the ray direction, respectively.

STGC-NeRF Overview

We now introduce STGC-NeRF, a LiDAR NeRF framework, which enhances dynamic scene reconstruction by incorporating spatial-temporal geometry consistency. The overall network architecture, as shown in Fig. 2, contains two sub-networks: a NeRF network for LiDAR point cloud rendering and a pre-trained scene flow network for flow estimation. STGC-NeRF mainly consists of two key components: (1) temporal geometric consistency regularization (TGCR) for improving the time-varying scene geometry regression from low-frequency LiDAR sequences, and (2) spatial geometric consistency constraint (SGCC) for enhancing local reconstruction details from sparse LiDAR data. We formulate the two components into end-to-end learning. During inference, only the NeRF network is required.

Temporal Geometric Consistency Regularization

Moving objects in dynamic scenes break the potential consistency between multiple training views, leading to inaccurate NeRFs rendering (Li et al. 2021, 2023). Especially for LiDAR point clouds, the low frequency (10-20 Hz) will lead to a notable displacement of dynamic objects, even in two consecutive frames. A recent method LiDAR4D (Zheng et al. 2024) employs an autoregressive scene flow network (Zheng et al. 2023) to provide motion priors. However, this flow is obtained by interpolating single-frame features, which is inaccurate for large displacements. As the common scene flow estimation network can directly predict flows at two moments as a geometric prior, incorporating it into NeRFs to enhance dynamic scene representation is natural. This paper proposes novel Temporal Geometric Consistency Regularization (TGCR), which explicitly converts scene flow from an off-the-shelf network into temporal constraints and improves dynamic scene representations.



(a) Flow from synthesized to real (b) Flow by autoregression

(c) Flow from real to real (d) enlarged details

Figure 3: Scene flow conceptual visualizations for TGCR. The yellow one is the point cloud at time t, and the blue one is the point cloud at time $t+i$. Arrows represent scene flow.

First, we align the movement of the synthesized LiDAR frames with that of the actual frames. Specifically, we compute the scene flow $\triangle \mathbf{F}^{syn\to r}_{t\to t\pm i}$ ($\triangle \mathbf{F}^{r\to r}_{t\to t\pm i}$) from the synthesized (real) point cloud at time t to the real one at $t \pm i$, as shown in Fig. 3 (a) and (c). Then, we minimize the discrepancy between $\triangle \mathbf{F}^{syn\to r}_{t\to t\pm i}$ and $\triangle \mathbf{F}^{r\to r}_{t\to t\pm i}$ to match the corresponding points in adjacent frames. For more effective regularization, we employ a term of cycle consistency to encourage both forward $\triangle \mathbf{F}^{syn\to r}_{t\to t\pm i}$ and backward $\triangle \mathbf{F}^{r\to syn}_{t\pm i\to t}$ flows to be consistent. The regularization $\mathcal{L}_{R1}$ for the first temporal supervision is formulated as:

$$\begin{aligned}\mathcal{L}_{R1} =& ||\triangle \mathbf{F}^{syn\to r}_{t\to t\pm i} - \triangle \mathbf{F}^{r\to r}_{t\to t\pm i}||_1 + \\ & ||\triangle \mathbf{F}^{r\to syn}_{t\pm i\to t} - \triangle \mathbf{F}^{r\to r}_{t\pm i\to t}||_1.\end{aligned} \tag{3}$$

Scene flows between adjacent frames offer accurate pointwise correspondences, enhancing the temporal coherence of synthesized views by adhering to actual motion states.

Second, we use the scene flow $\triangle \mathbf{F}^{r\to r}_{t\to t\pm i}$ from actual sequences as additional supervision to enhance dynamic representations. Following LiDAR4D, we employ an autoregressive flow MLP network (Fig. 2) to estimate scene flow $\triangle \mathbf{F}^{interp}_{t\to t\pm i}$ by single-frame feature interpolation with the given timestamp, as shown in Fig. 3 (b). However, unlike LiDAR4D, we supervise $\triangle \mathbf{F}^{interp}_{t\to t\pm i}$ by $\triangle \mathbf{F}^{r\to r}_{t\to t\pm i}$, which explicitly leverage correlations between LiDAR sequences. In addition, the network is trained with a Chamfer Distance loss $loss_{CD} = CD(S_t + \triangle \mathbf{F}^{interp}_{t\to t\pm i}, S_{t+i})$. The regularization $\mathcal{L}_{R2}$ for the second temporal supervision is depicted as:

$$\mathcal{L}_{R2} = ||\triangle \mathbf{F}^{interp}_{t\to t\pm i} - \triangle \mathbf{F}^{r\to r}_{t\to t\pm i}||_1 + loss_{CD}. \tag{4}$$

Since the flow $\triangle \mathbf{F}^{r\to r}_{t\to t\pm i}$ provides more accurate motion estimation, we can leverage it to supervise the autoregressive flow for enhanced motion prior.

Finally, the total TGCR ($\mathcal{L}_T$) is comprised of the two aforementioned regularizations, defined as follows:

$$\mathcal{L}_T = \alpha_1 \mathcal{L}_{R1} + \alpha_2 \mathcal{L}_{R2}, \tag{5}$$

which can effectively regularize the dynamic scene representation for more accurate reconstruction. In this paper, we employ GMSF (Zhang et al. 2023), a matching-based scene flow estimation network, for flow prediction. The GMSF network is frozen and will be dropped during inference.

Spatial Geometric Consistency Constraint

Due to the sparse characteristics of the LiDAR data, the point cloud will be less dense the further from the sensor. This will result in inconsistencies across consecutive training views, accompanied by a loss of local detail. Current LiDAR NeRFs (Tao et al. 2023; Uy et al. 2023; Zheng et al. 2024) only minimize the individual point discrepancy during training, which ignores the local neighborhood relationship between the points. The reconstruction will be prone to a lack of local structural details, e.g., voids or faults. To solve this problem, we propose a novel Spatial Geometric Consistency Constraint (SGCC) to improve the reconstruction details of sparse LiDAR data. We compute geometric and contextual feature descriptors for neighborhood points. Then, we convert them into spatial supervision.

Specifically, we categorize the features suitable for describing local scene structures into two classes: fundamental geometric features and contextual features $\mathbf{F}_c$. The former includes eigenvalue/eigenvector-based features $\mathbf{F}_e$ and height features along the z-axis $\mathbf{F}_z$. We construct local point sets $\mathcal{N}(\mathbf{s}_i)$ with k nearest neighboring points for each point $\mathbf{s}_i$. Then, we compute the local covariance matrix and obtain the eigenvalues $\lambda_1 \le \lambda_2 \le \lambda_3$ and eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ by eigendecomposition. Meanwhile, we use the GMSF feature encoder (the scene flow network) and a VLAD network (Uy and Lee 2018) (commonly used global context description) to extract local $\mathbf{f^l}$ and global $\mathbf{f^g}$ features, respectively. We can describe $\mathbf{F}_e$, $\mathbf{F}_z$, and $\mathbf{F}_c$ as:

- $\mathbf{F}_e$ features: Normal vector $\mathbf{n}_i = \mathbf{v}_1$, Curvature $curv_i = \frac{\lambda_1}{\lambda_1+\lambda_2+\lambda_3}$, and Point density $\rho_i = \frac{\sum_{\mathbf{s}_j \in \mathcal{N}(\mathbf{s}_i)} \|\mathbf{s}_i - \mathbf{s}_j\|_2^2}{k}$.
- $\mathbf{F}_z$ features: Maximum height difference $\Delta h = \max_{1\le i\le k}(z_i) - \min_{1\le i\le k}(z_i)$ and Height variance $\sigma_h^2 = \frac{1}{k}\sum_{i=1}^{k}(z_i - \frac{1}{k}\sum_{j=1}^{k} z_j)^2$.
- $\mathbf{F}_c$ features: local context feature $\mathbf{f^l}(\mathcal{N}(\mathbf{s}_i))$ and global context feature $\mathbf{f^g}(\{\mathbf{s}_i\}_{i=1}^N)$.

The complete SGCC ($\mathcal{L}_S$) consists of constraints on three features, which can be depicted as:

$$\mathcal{L}_S = \beta_1||\hat{\mathbf{F}}_e - \mathbf{F}_e||_1 + \beta_2||\hat{\mathbf{F}}_z - \mathbf{F}_z||_1 + \beta_3||\hat{\mathbf{F}}_c - \mathbf{F}_c||_2, \tag{6}$$

where $\hat{\mathbf{F}}$ and $\mathbf{F}$ are features obtained from synthesized and ground-truth point clouds. $\{\beta_i\}_{i=1}^3$ are weight coefficients to balance different feature terms. For the proposed SGCC, the feature $\mathbf{F}_e$ ensures the consistency of local surface and spatial distribution. $\mathbf{F}_z$ additionally captures vertical structure consistency and height distributions. $\mathbf{F}_c$ preserves detailed contextual information for each point and the scene. By integrating these spatial feature constraints, the model can better capture the 3D geometric details, leading to more accurate local reconstructed surfaces with consistent direction, smoothness, and detail.

Loss Function

Following conventional LiDAR NeRFs (Tao et al. 2023; Huang et al. 2023), the optimization includes three objectives: depth loss, intensity loss, and ray-drop loss. The depth loss $\mathcal{L}_D = \frac{1}{|\mathcal{R}|}\sum_{\mathbf{r}\in\mathcal{R}} ||\hat{D}(\mathbf{r}) - D(\mathbf{r})||_1$ and the intensity

Method	Type	Point Cloud		Depth					Intensity				
		CD↓	F-score↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑
LiDARsim	$\mathcal{E}./\mathcal{S}./\mathcal{M}.$	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	$\mathcal{E}./\mathcal{S}./\mathcal{M}.$	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	$\mathcal{E}./\mathcal{S}.$	0.4636	0.8023	5.6583	0.2040	0.5391	0.4903	23.1675	0.1970	0.0763	0.5926	0.1351	14.1181
LiDAR-NeRF	$\mathcal{I}./\mathcal{S}.$	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	$\mathcal{I}./\mathcal{D}.$	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	$\mathcal{I}./\mathcal{D}.$	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	$\mathcal{I}./\mathcal{D}.$	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	$\mathcal{I}./\mathcal{D}.$	0.1089	0.9272	3.5256	0.0404	0.1051	0.7647	27.4767	0.1195	0.0327	0.1845	0.5304	18.5561
STGC-NeRF	$\mathcal{I}./\mathcal{D}.$	**0.0997**	**0.9325**	**3.0794**	**0.0277**	**0.0681**	**0.8774**	**28.6796**	**0.0995**	**0.0262**	**0.1479**	**0.6563**	**20.0825**

Table 1: Comparison with state-of-the-art methods on the **KITTI-360** dataset. The best result is in **bold** and the second best is underlined. $\mathcal{E}$: Explicit, $\mathcal{I}$: Implicit, $\mathcal{S}$: Static, $\mathcal{D}$: Dynamic, $\mathcal{M}$: Mesh.

Method	Type	Point Cloud		Depth					Intensity				
		CD↓	F-score↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑
LiDARsim	$\mathcal{E}./\mathcal{S}./\mathcal{M}.$	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	$\mathcal{E}./\mathcal{S}./\mathcal{M}.$	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	$\mathcal{E}./\mathcal{S}.$	2.1998	0.6341	8.8364	0.4011	0.1792	0.5440	19.2799	0.0768	0.0147	0.1308	0.4410	22.4428
LiDAR-NeRF	$\mathcal{I}./\mathcal{S}.$	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	$\mathcal{I}./\mathcal{D}.$	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	$\mathcal{I}./\mathcal{D}.$	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	$\mathcal{I}./\mathcal{D}.$	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	$\mathcal{I}./\mathcal{D}.$	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	$\mathcal{I}./\mathcal{D}.$	**0.2204**	**0.9070**	**6.5361**	**0.0240**	**0.0486**	**0.7741**	**22.0044**	**0.0417**	0.0082	**0.0418**	**0.7566**	**27.9989**

Table 2: Comparison with state-of-the-art methods on the **nuScenes** dataset. The notations are consistent with Tab. 1.

loss $\mathcal{L}_I = \frac{1}{|\mathcal{R}|}\sum_{\mathbf{r}\in\mathcal{R}} ||\hat{I}(\mathbf{r}) - I(\mathbf{r})||_2^2$ are used to reconstruct the depth and intensity of the LiDAR point clouds, respectively. $\mathcal{R}$ is the set of LiDAR rays. The ray-drop loss $\mathcal{L}_P = \frac{1}{|\mathcal{R}|}\sum_{\mathbf{r}\in\mathcal{R}} ||\hat{P}(\mathbf{r}) - P(\mathbf{r})||_2^2$ is to simulate the emitted rays that are not reflected back to the sensor. In addition, we employ the ray-drop refinement loss $\mathcal{L}_R = \mathcal{L}_{bce}(\hat{\mathcal{M}}, \mathcal{M})$ from (Zheng et al. 2024) to refine the final ray-drop mask via a binary cross entropy loss $\mathcal{L}_{bce}$. $\hat{\mathcal{M}}$ and $\mathcal{M}$ are rendered and ground-truth masks, respectively. Finally, by incorporating our proposed TGCR ($\mathcal{L}_T$) and SGCC ($\mathcal{L}_S$), the overall loss function is defined as:

$$\mathcal{L} = \lambda_1\mathcal{L}_D + \lambda_2\mathcal{L}_I + \lambda_3\mathcal{L}_P + \lambda_4\mathcal{L}_R + \lambda_5\mathcal{L}_T + \lambda_6\mathcal{L}_S, \quad (7)$$

where $\{\lambda_i\}_{i=1}^6$ are weights to balance different loss terms.

Experiments

Experimental Settings

Datasets. We conduct experiments on two challenging autonomous driving datasets: KITTI-360 (Liao, Xie, and Geiger 2022) and nuScenes (Caesar et al. 2020). Each dataset consists of a large amount of dynamic objects. KITTI-360 is captured by a Velodyne HDL-64E LiDAR sensor (-24.8°to 2°vertical FOV) at 10 Hz. nuScenes is collected by a Velodyne HDL-32E LiDAR sensor (-30°to 10°vertical FOV) at 20 Hz. (1) Following (Zheng et al. 2024), we construct 6 dynamic scenes from KITTI-360 and 5 dynamic scenes from nuScenes for evaluation. Each dynamic scene contains 51 consecutive frames with 4 equidistant test samples. (2) We also employ 4 static scenes from KITTI-360 to evaluate our method as introduced in (Tao et al. 2023). Each static scene comprises 64 consecutive frames with 4 equidistant test samples.

Implementation Details. We implement our STGC-NeRF using LiDAR4D (Zheng et al. 2024) as the backbone NeRF network. The proposed method is implemented with PyTorch (Paszke et al. 2019) using an Adam optimizer and an initial learning rate of 0.01. Each scene is trained in 30k iterations with a batch size of 1024 rays on a single NVIDIA RTX 4090 GPU. In Eq. 5, α_1 is 0.1 and α_2 is 1. In Eq. 6, β_1 and β_2 are 0.1, and β_3 is 1. In Eq. 7, λ_2 is 0.1, λ_3 is 0.01, and all other λ are set to 1. k is 12. Consistent with previous LiDAR NeRFs (Tao et al. 2023; Zheng et al. 2024), we convert point clouds into range images for training. For scene flow estimation, we employ pre-trained GMSF (Zhang et al. 2023), loading weights trained on the KITTI Scene Flow dataset (Menze and Geiger 2015). Then, we directly perform scene flow inference for our training process.

Baselines. We compare our STGC-NeRF with two types of baselines: explicit reconstruction methods and implicit NeRFs methods. LiDARsim (Manivasagam et al. 2020), NKSR (Huang et al. 2023), and PCGen (Li, Ren, and Liu 2023) are explicit baselines. We follow (Zheng et al. 2024) to migrate dynamic NeRFs, D-NeRF (Pumarola et al. 2021), TiNeuVox (Fang et al. 2022), and K-Planes (Fridovich-Keil et al. 2023), to LiDAR scene reconstruction for a more comprehensive comparison. LiDAR-NeRF (Tao et al. 2023) and LiDAR4D (Zheng et al. 2024) are primary comparison

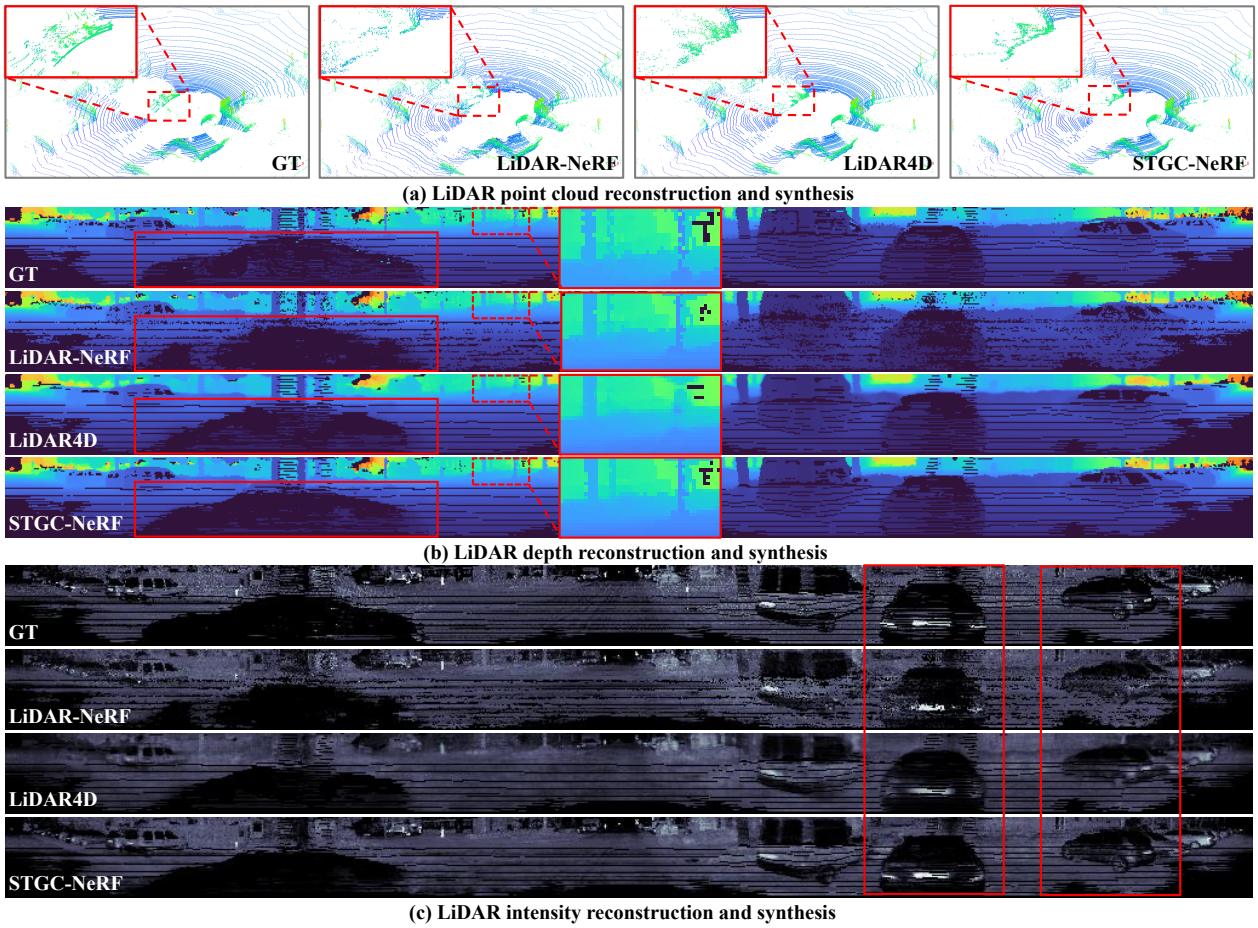


Figure 4: Qualitative evaluation of rendering **LiDAR point cloud**, **LiDAR depth**, and **LiDAR intensity** on the KITTI-360 dataset. Our STGC-NeRF estimates more accurate results, especially within the red boxes.

methods. They are both LiDAR NeRFs, whereas LiDAR4D further focuses on dynamic scene reconstruction.

Metrics. For a comprehensive evaluation, the Chamfer distance (CD) and the F-score value (error threshold of 5cm) are first reported to evaluate the reconstruction accuracy of LiDAR point clouds. To evaluate reconstructed depth and intensity, we use the root mean square error (RMSE) and median absolute error (MedAE) to evaluate the pixel-wise error of range images. LPIPS (Zhang et al. 2018), SSIM (Wang et al. 2004), and PSNR are also employed to measure the overall variance for range images.

Comparison with State-of-the-art Methods

KITTI360 Results. In Tab. 1, we first report the evaluation of the KITTI-360 dataset. For point cloud reconstruction accuracy, the proposed STGC-NeRF achieves a CD error of 0.0997, outperforming all competitors. Specifically, compared to the state-of-the-art method LiDAR4D, it improves by 8.4% on the CD error. In addition, the evaluation metrics, for depth RMSE and intensity PSNR, are also ahead of comparison methods, improving by 12.7% and 8.2%, respectively. These results demonstrate the effectiveness of our method in both geometry and intensity accuracy for dynamic scene reconstruction. The notable improvement is due to the proposed spatial-temporal geometric consistency. In addition, we follow the method LiDAR-NeRF to evaluate our STGC-NeRF on KITTI-360 static scenes. The results in Tab. 3 still demonstrate the effectiveness of our method.

Fig. 4 presents the visualization on the KITTI-360 dataset. For both point cloud, depth, and intensity reconstruction, the qualitative results of STGC-NeRF are all closer to the ground truth than the competitors. LiDAR-NeRF is a static NeRF method, resulting in significant discrete noise on dynamic regions. Although LiDAR4D uses flow autoregression for motion priors, the flow is not accurate due to the large object motion, resulting in obvious blurring. The visualizations still demonstrate that STGC-NeRF can perform better dynamic scene reconstruction than others.

nuScenes Results. We further report the comparison results on the nuScenes dataset in Tab. 2. Our method achieves 0.2204 average CD error, which ranks first compared to other methods, achieving the smallest CD error. Compared to LiDAR4D, whose result is 0.2443, STGC-NeRF shows a 9.8% significant improvement in point cloud synthesis. Fur-

Method	Type	Point Cloud		Depth					Intensity				
		CD↓	F-score↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑
LiDARsim	$\mathcal{E}./\mathcal{S}./\mathcal{M}.$	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	$\mathcal{E}./\mathcal{S}./\mathcal{M}.$	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	$\mathcal{E}./\mathcal{S}.$	0.2090	0.8597	4.8838	0.1785	0.5210	0.5062	24.3050	0.2005	0.0818	0.6100	0.1248	13.9606
LiDAR-NeRF	$\mathcal{I}./\mathcal{S}.$	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	$\mathcal{I}./\mathcal{D}.$	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	$\mathcal{I}./\mathcal{D}.$	**0.0831**	**0.9332**	**2.7717**	**0.0353**	**0.0985**	**0.8480**	**29.2388**	**0.1121**	**0.0336**	**0.1822**	**0.6116**	**19.0238**

Table 3: Comparison with state-of-the-art methods on **KITTI-360 Static Scenes**. The notations are consistent with Tab. 1.

	TGCR		SGCC			Point Cloud		Depth					Intensity				
	$\mathcal{L}_{R1}$	$\mathcal{L}_{R2}$	$\mathbf{F}_e$	$\mathbf{F}_z$	$\mathbf{F}_c$	CD↓	F-score↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑	RMSE↓	MedAE↓	LPIPS↓	SSIM↑	PSNR↑
1						0.1089	0.9272	3.5256	0.0404	0.1051	0.7647	27.4767	0.1195	0.0327	0.1845	0.5304	18.5561
2	✓					0.1033	0.9291	3.2660	0.0338	0.0834	0.8347	27.9228	0.1044	0.0294	0.1689	0.6080	19.2568
3		✓				0.1039	0.9292	3.2470	0.0369	0.0826	0.8369	27.9881	0.1054	0.0292	0.1669	0.6057	19.1790
4	✓	✓				0.1019	0.9301	3.1624	0.0298	0.0798	0.8585	28.2323	0.1023	0.0283	0.1597	0.6263	19.5132
5			✓			0.1038	0.9286	3.2619	0.0368	0.0851	0.8255	27.8460	0.1033	0.0304	0.1698	0.6041	19.2519
6				✓		0.1047	0.9274	3.3146	0.0398	0.0854	0.8274	27.7282	0.1054	0.0319	0.1712	0.5994	19.0519
7					✓	0.1025	0.9305	3.2133	0.0340	0.0803	0.8339	28.0672	0.1024	0.0307	0.1627	0.6184	19.2455
8			✓	✓	✓	0.1011	0.9313	3.1644	0.0299	0.0786	0.8573	28.3369	0.1015	0.0272	0.1590	0.6168	19.6032
9	✓	✓	✓	✓	✓	0.0997	0.9325	3.0794	0.0277	0.0681	0.8774	28.6796	0.0995	0.0262	0.1479	0.6563	20.0825

Table 4: Ablation study of **TGCR** and **SGCC** on the KITTI-360 dataset. $\mathcal{L}_{R1}$ and $\mathcal{L}_{R2}$ are the first and second regularization in TGCR. Fundamental geometric features $\mathbf{F}_e$/$\mathbf{F}_z$ and contextual features $\mathbf{F}_c$ are all belong to SGCC. Row 1 is the vanilla model without any proposed modules. Row 9 is the complete STGC-NeRF.

ther, STGC-NeRF also obtains the best performance across most metrics. In the nuScenes dataset, there are also a large number of dynamic objects as in the KITTI-360 dataset. Therefore, our method achieves high performance in dynamic scene reconstruction by effectively enforcing geometric consistency across temporal and spatial dimensions.

Ablation Study

Ablation of Temporal Geometric Consistency Regularization (TGCR). We mainly report the ablation results on the KITTI-360 dataset. As shown in Tab. 4, we first perform ablation studies of the proposed TGCR. Only using regularization $\mathcal{L}_{R1}$ (Row 2) improves the depth RMSE by 7.4% compared to the vanilla model (Row 1). It shows the effectiveness of $\mathcal{L}_{R1}$ in regularizing temporal coherence by scene flow supervision. The comparison between Row 3, only employing $\mathcal{L}_{R2}$, and Row 1 reveals that $\mathcal{L}_{R2}$ achieves an average improvement of 7.9% in depth RMSE. This demonstrates that $\mathcal{L}_{R2}$ can be an effective form of motion priors in dynamic scene representation. Compared to Row 1, STGC-NeRF with total TGCR (Row 4) outperforms the vanilla model by 10.3%/5.2% on depth RMSE and intensity PSNR. This indicates that TGCR can improve dynamic reconstruction accuracy by enhancing the regression of time-varying scene geometries from low-frequency LiDAR sequences.

Ablation of Spatial Geometric Consistency Constraint (SGCC). The ablation experiments of the proposed SGCC are reported in Tab. 4. In Rows 5, 6, and 7, we evaluate the impact of fundamental geometric features ($\mathbf{F}_e$ and $\mathbf{F}_z$) and contextual features ($\mathbf{F}_c$). Using $\mathbf{F}_e$ improves depth RMSE better than Row 1, an average improvement of 7.5%, as it ensures local surface and spatial consistency. Employing $\mathbf{F}_z$ also shows a better improvement in depth RMSE (6.0%), as it captures the consistency and distribution of the vertical structure. Since $\mathbf{F}_c$ preserves detailed local and global contextual information, it gains a 8.9% increase on the above metric. Finally, the comparison between Row 8 and Row 1 shows that SGCC obtains an average improvement of 10.2%/5.6% (depth RMSE/intensity PSNR). All these results demonstrate that SGCC is effective in improving the local reconstruction details with sparse LiDAR data.

Conclusion

In this paper, we address the challenge of learning dynamic scene representations in LiDAR NeRFs, prompted by the low frequency and sparsity of LiDAR point clouds. These are the most important reasons for the performance gap between other LiDAR NeRFs and ours in dynamic scene reconstruction. We propose a novel framework, STGC-NeRF, which regularizes LiDAR NeRFs across spatial-temporal geometric consistency. Specifically, we propose temporal geometric consistency regularization, improving the regression of time-varying scene geometries from low-frequency LiDAR sequences. To enhance the local reconstruction details of sparse LiDAR point clouds, we propose spatial geometric consistency constraints to integrate neighborhood geometric and contextual constraints. Extensive experiments demonstrate the effectiveness of our method.

References

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 5855–5864.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 5470–5479.

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 *CVPR*, 11621–11631.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *ECCV*, 333–350.

Dong, S.; Xu, K.; Zhou, Q.; Tagliasacchi, A.; Xin, S.; Nießner, M.; and Chen, B. 2019. Multi-robot collaborative dense scene reconstruction. *TOG*, 38(4): 1–16.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *CoRL*, 1–16.

Fang, J.; Yi, T.; Wang, X.; Xie, L.; Zhang, X.; Liu, W.; Nießner, M.; and Tian, Q. 2022. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia*, 1–9.

Fang, J.; Zhou, D.; Yan, F.; Zhao, T.; Zhang, F.; Ma, Y.; Wang, L.; and Yang, R. 2020. Augmented LiDAR simulator for autonomous driving. *RA-L*, 5(2): 1931–1938.

Fang, X.; Fang, W.; Liu, D.; Qu, X.; Dong, J.; Zhou, P.; Li, R.; Xu, Z.; Chen, L.; Zheng, P.; et al. 2024a. Not all inputs are valid: Towards open-set video moment retrieval using language. In *ACM MM*, 28–37.

Fang, X.; Liu, D.; Fang, W.; Zhou, P.; Xu, Z.; Xu, W.; Chen, J.; and Li, R. 2024b. Fewer Steps, Better Performance: Efficient Cross-Modal Clip Trimming for Video Moment Retrieval Using Language. In *AAAI*, 1735–1743.

Fang, X.; Liu, D.; Zhou, P.; and Nan, G. 2023. You can ground earlier than see: An effective and efficient pipeline for temporal sentence grounding in compressed videos. In *CVPR*, 2448–2460.

Fridovich-Keil, S.; Meanti, G.; Warburg, F. R.; Recht, B.; and Kanazawa, A. 2023. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 12479–12488.

Gschwandtner, M.; Kwitt, R.; Uhl, A.; and Pree, W. 2011. BlenSor: Blender sensor simulation toolbox. In *Advances in Visual Computing*, 199–208.

Hu, W.; Wang, Y.; Ma, L.; Yang, B.; Gao, L.; Liu, X.; and Ma, Y. 2023. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *ICCV*, 19774–19783.

Hu, X.; Xiong, G.; Zang, Z.; Jia, P.; Han, Y.; and Ma, J. 2024. PC-NeRF: Parent-Child Neural Radiance Fields Using Sparse LiDAR Frames in Autonomous Driving Environments. *TIV*, 1–14.

Huang, S.; Gojcic, Z.; Wang, Z.; Williams, F.; Kasten, Y.; Fidler, S.; Schindler, K.; and Litany, O. 2023. Neural lidar fields for novel view synthesis. In *ICCV*, 18236–18246.

Kniaz, V.; Knyaz, V.; Bordodymov, A.; Moshkantsev, P.; Novikov, D.; and Barylnik, S. 2023. Double Nerf: Representing Dynamic Scenes as Neural Radiance Fields. *ISPRS Archives*, 48: 115–120.

Koenig, N.; and Howard, A. 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IROS*, volume 3, 2149–2154.

Li, C.; Ren, Y.; and Liu, B. 2023. Pcgen: Point cloud generator for lidar simulation. In *ICRA*, 11676–11682.

Li, W.; Yang, Y.; Yu, S.; Hu, G.; Wen, C.; Cheng, M.; and Wang, C. 2024. DiffLoc: Diffusion Model for Outdoor LiDAR Localization. In *CVPR*, 15045–15054.

Li, Z.; Niklaus, S.; Snavely, N.; and Wang, O. 2021. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 6498–6508.

Li, Z.; Wang, Q.; Cole, F.; Tucker, R.; and Snavely, N. 2023. Dynibar: Neural dynamic image-based rendering. In *CVPR*, 4273–4284.

Liao, Y.; Xie, J.; and Geiger, A. 2022. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *TPAMI*, 45(3): 3292–3310.

Liu, L.; Gu, J.; Zaw Lin, K.; Chua, T.-S.; and Theobalt, C. 2020. Neural sparse voxel fields. *NeurIPS*, 33: 15651–15663.

Liu, Y.-L.; Gao, C.; Meuleman, A.; Tseng, H.-Y.; Saraf, A.; Kim, C.; Chuang, Y.-Y.; Kopf, J.; and Huang, J.-B. 2023. Robust dynamic radiance fields. In *CVPR*, 13–23.

Manivasagam, S.; Wang, S.; Wong, K.; Zeng, W.; Sazanovich, M.; Tan, S.; Yang, B.; Ma, W.-C.; and Urtasun, R. 2020. Lidarsim: Realistic lidar simulation by leveraging the real world. In *CVPR*, 11164–11173.

Menze, M.; and Geiger, A. 2015. Object scene flow for autonomous vehicles. In *ICCV*, 3061–3070.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 41(4): 1–15.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, volume 32.

Pfister, H.; Zwicker, M.; Van Baar, J.; and Gross, M. 2000. Surfels: Surface elements as rendering primitives. In *SIGGRAPH*, 335–342.

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 10318–10327.

Shah, S.; Dey, D.; Lovett, C.; and Kapoor, A. 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *FSR*, 621–635.

Sun, S.; Zhuang, B.; Jiang, Z.; Liu, B.; Xie, X.; and Chandraker, M. 2024a. LidaRF: Delving into Lidar for Neural Radiance Field on Street Scenes. In *CVPR*, 19563–19572.

Sun, X.; Xu, Q.; Yang, X.; Zang, Y.; and Wang, C. 2024b. Global and Hierarchical Geometry Consistency Priors for Few-shot NeRFs in Indoor Scenes. In *CVPR*, 20530–20539.

Tao, T.; Gao, L.; Wang, G.; Lao, Y.; Chen, P.; Zhao, H.; Hao, D.; Liang, X.; Salzmann, M.; and Yu, K. 2023. LiDAR-NeRF: Novel lidar view synthesis via neural radiance fields. *arXiv preprint arXiv:2304.10406*.

Tao, T.; Wang, G.; Lao, Y.; Chen, P.; Liu, J.; Lin, L.; Yu, K.; and Liang, X. 2024. AlignMiF: Geometry-Aligned Multimodal Implicit Field for LiDAR-Camera Joint Synthesis. In *CVPR*, 21230–21240.

Uy, M. A.; and Lee, G. H. 2018. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *CVPR*, 4470–4479.

Uy, M. A.; Martin-Brualla, R.; Guibas, L.; and Li, K. 2023. Scade: Nerfs from space carving with ambiguity-aware depth estimates. In *CVPR*, 16518–16527.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4): 600–612.

Wang, Z.; Shen, T.; Gao, J.; Huang, S.; Munkberg, J.; Hasselgren, J.; Gojcic, Z.; Chen, W.; and Fidler, S. 2023. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *CVPR*, 8370–8380.

Wu, H.; Zuo, X.; Leutenegger, S.; Litany, O.; Schindler, K.; and Huang, S. 2024. Dynamic LiDAR Re-simulation using Compositional Neural Fields. In *CVPR*, 19988–19998.

Yan, Z.; Li, C.; and Lee, G. H. 2023. Nerf-ds: Neural radiance fields for dynamic specular objects. In *CVPR*, 8285–8295.

Yang, Z.; Chen, Y.; Wang, J.; Manivasagam, S.; Ma, W.-C.; Yang, A. J.; and Urtasun, R. 2023. Unisim: A neural closed-loop sensor simulator. In *CVPR*, 1389–1399.

Yu, S.; Sun, X.; Li, W.; Wen, C.; Yang, Y.; Si, B.; Hu, G.; and Wang, C. 2024. NIDALoc: Neurobiologically Inspired Deep LiDAR Localization. *TITS*, 25(5): 4278–4289.

Zhang, J.; Zhang, F.; Kuang, S.; and Zhang, L. 2024. Nerf-lidar: Generating realistic lidar point clouds with neural radiance fields. In *AAAI*, volume 38, 7178–7186.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *ICCV*, 586–595.

Zhang, Y.; Edstedt, J.; Wandt, B.; Forssen, P.-E.; Magnusson, M.; and Felsberg, M. 2023. GMSF: Global Matching Scene Flow. In *NeurIPS*, volume 36, 64415–64427.

Zheng, Z.; Lu, F.; Xue, W.; Chen, G.; and Jiang, C. 2024. LiDAR4D: Dynamic Neural Fields for Novel Space-time View LiDAR Synthesis. In *CVPR*, 5145–5154.

Zheng, Z.; Wu, D.; Lu, R.; Lu, F.; Chen, G.; and Jiang, C. 2023. Neuralpci: Spatio-temporal neural field for 3d point cloud multi-frame non-linear interpolation. In *CVPR*, 909–918.