Neural LiDAR Fields for Novel View Synthesis

Shengyu Huang[1,2] Zan Gojcic[2] Zian Wang[2,3,4] Francis Williams[2]
Yoni Kasten[2] Sanja Fidler[2,3,4] Konrad Schindler[1] Or Litany[2]
[1] ETH Zurich [2] NVIDIA [3] University of Toronto [4] Vector Institute
https://research.nvidia.com/labs/toronto-ai/nfl/

Abstract

We present Neural Fields for LiDAR (NFL), a method to optimise a neural field scene representation from LiDAR measurements, with the goal of synthesizing realistic LiDAR scans from novel viewpoints. NFL combines the rendering power of neural fields with a detailed, physically motivated model of the LiDAR sensing process, thus enabling it to accurately reproduce key sensor behaviors like beam divergence, secondary returns, and ray dropping. We evaluate NFL on synthetic and real LiDAR scans and show that it outperforms explicit reconstruct-then-simulate methods as well as other NeRF-style methods on LiDAR novel view synthesis task. Moreover, we show that the improved realism of the synthesized views narrows the domain gap to real scans and translates to better registration and semantic segmentation performance.

1. Introduction

The goal of novel view synthesis is to generate a view of a 3D scene, from a viewpoint at which no real sensor image has been captured. This offers the possibility to observe *real* scenes from a *virtual*, unobserved perspective. Among other applications, it has tremendous potential for autonomous driving: synthetic novel views may be used to train and test perception algorithms across a wider range of viewing conditions, thus enhancing robustness and generalization. Moreover, novel view synthesis becomes critical when the desired viewpoints are not known in advance, *e.g.*, during training of a planning module whose decisions determine future vehicle locations.

Neural radiance fields (NeRFs) have led to unprecedented visual quality when synthesizing novel camera views [2, 29, 30, 52]. These methods represent the 3D scene in form of continuous density and radiance fields, from which images can be generated through volume rendering, mimicking the image acquisition process. The inductive bias of neural networks imparts NeRFs the ability to interpolate complex lighting and reflectance behaviours with a high degree of realism.

While most prior works focused on synthesizing camera views, 3D perception in the autonomous driving context typically relies partly (or even exclusively) on LiDAR measurements. Synthesizing realistic LiDAR scans from novel viewpoints thus has a lot of potential for data augmentation and closed-loop testing of autonomous navigation systems.

The problem of synthesizing novel LiDAR views has previously been addressed in two stages [24]. First, extract an explicit surface representation such as surfels or a triangular mesh from the scanned point clouds. Then, simulate LiDAR measurements from a novel viewpoint by casting rays and intersecting them with the surface model. Like for images, explicit reconstruction (which is not optimised towards the subsequent synthesis step) suffers from discretization artifacts and introduces noticeable errors [46]. Moreover, the rendering assumes an idealised ray model and neglects the divergence of the LiDAR beams, which causes frequent second returns from distant surfaces.

Here, we instead build on a main insight of NeRF [29]: directly optimizing an implicit scene representation for novel view synthesis can produce more realistic outputs than the reconstruct-then-simulate approach. Specifically, we propose Neural Fields for LiDAR (NFL), a NeRF-style representation for synthesizing novel LiDAR viewpoints.

Several NeRF extensions have utilized range measurements as additional supervision, and have shown that constraining the scene geometry more tightly can yield better (camera) view synthesis [8, 38]. Yet, the output of those methods are synthetic images, not LiDAR scans, consequently they have not paid attention to effects specific to LiDAR sensing: a laser scanner does *not* directly sense range, rather it measures the returned light energy per ray and determines the range based on the waveform. This includes the possibilities that there are multiple returns[1] from the emitted ray, or no return at all.

Our formulation closely adheres to the principles of the LiDAR measurement process and incorporates them into

[1]In principle there can be >2 returns, but automotive LiDAR sensors typically record the first two echos.

the neural field framework. Specifically, we (i) **devise volume rendering for LiDAR sensors**; (ii) **incorporate beam divergence** and (iii) **propose truncated volume rendering** to account for secondary returns and improve range prediction.

We evaluate our method on both synthetic and real LiDAR data. To this end, we (iv) **develop a LiDAR simulator** for synthesizing scenes from 3D assets that serve as a test bed for viewpoints far from the original scan locations, and to study the effect of different scan patterns. Real data from the Waymo [43] dataset is used to evaluate NFL against real scans at held-out viewpoints, including real-world intensities, ray drops and secondary returns. Additionally, we (v) **propose a novel closed-loop evaluation protocol** that leverages real data to evaluate view synthesis in challenging views. As an end-to-end test for downstream tasks, we further evaluate the performance of state-of-the-art segmentation and registration networks when trained on real scans and tested on novel views generated by NFL.

2. Related Work

LiDAR simulation. Simulating realistic LiDAR data is useful for training perception models. Different from real-world LiDAR data that requires annotation efforts, simulated data can be automatically generated with ground truth labels, *e.g*. object bounding boxes and semantic segmentation. Unrealistic LiDAR simulation will prevent the trained models from generalizing to real data. Traditional simulation engines, such as those proposed in [9, 19], require the specification of sensor parameters and 3D scene assets and use ray-casting methods for simulation. Although these point clouds can accurately represent scene geometry, they often exhibit a discrepancy, or "domain gap", compared to real data, due to the lack of modeling for sensor noise, such as ray drop and Gaussian beam. Furthermore, this approach relies heavily on the creation of 3D scene assets, which can be time-consuming and expensive. To address these challenges, LiDARsim [24] reconstructs the static and dynamic scene assets from real data using surfel [34] representation and models the ray-drop pattern for improved realism. BaiduSim [10] proposes a probability map to model scene compositions in order to reduce the domain gap. Most recent work [11] learns to enhance existing simulated LiDAR intensity and ray-drop patterns, using the available corresponding RGB images.

Weather conditions, such as fog or rain, can significantly impact the quality of LiDAR data, and downstream models trained solely on ideal weather conditions may fail to generalize to these effects. Recent methods and datasets [5, 12, 13, 17] have been proposed to address this issue. SnowSIM [12] and FogSIM [13] sample snow particles and model the impulse response from atmospheric attenuation, respectively, to alter the range measurements of each ray. Other approaches [17, 20, 42] simulate LiDAR data on rainy days and the spray effects, in a similar fashion.

NeRF for Novel View Synthesis. NeRF [29] maps 5D position and direction to density and radiance scene values, and uses volume rendering [25, 26] to estimate pixel color. This technique has proven effective for generating realistic images at unseen camera views. Many methods have been proposed to improve robustness to camera poses [7, 21], handle dynamics [32, 35], anti-alias [2, 3, 53], and speed up optimisation [22, 30, 52] *etc*. Despite its high-quality novel view synthesis capacity, the underlying geometry of NeRF is considered inaccurate and noisy [31], making it less favoured for geometry reconstruction, especially in sparse-views settings. [31, 48, 51] address this challenge by using implicit surface representations, and defining the density functions based on them to enabling volume rendering. DS-NeRF [8] and DenseDS-NeRF [39] use sparse depth supervision from SfM [41] points to regularise the density field. Urban Radiance Field [38] leverages LiDAR data for depth supervision.

Neural fields beyond regular cameras. Neural fields are a natural and continuous representation [50] for spatio-temporal information including SDFs [33], occupancy [27] and radiance field [29] *etc*. While in its original form, NeRF [29] performs novel view synthesis using tonemapped low dynamic range images, RawNeRF [28] extends it to operate over the high dynamic range images, enabling additional adjustments to focus, exposure, and tonemapping. Törf [1] incorporates the image formation model for continuous-wave Time-of-Flight (ToF) cameras into NeRF, allowing it to jointly process RGB and ToF sensor data and improve reconstruction robustness to large motions. EventNeRF [40] and ENeRF [18] optimise the scene representation for Novel View Synthesis (NVS) from sparse event streams that contain asynchronous per-pixel brightness change signals. Other works [23, 36] explore the use of acoustic signals for surface reconstruction or NVS.

3. Background

We start by reviewing the principles of volume rendering (Sec. 3.1) and the sensor model for LiDAR (Sec. 3.2). This sets the stage for the proposed formulation of Neural LiDAR Fields (Sec. 4).

3.1. Volume rendering for passive sensors

In the following, we provide a brief summary of camera-based volume rendering as used by NeRF [29, 44]. This will serve as the basis to derive volume rendering equations for the active LiDAR sensor.

Density and transmittance. For a ray $\mathbf{r}(\mathbf{o}, \mathbf{d})$ emitted from the origin $\mathbf{o} \in \mathbb{R}^3$ in direction $\mathbf{d} \in \mathbb{R}^3$, the *density*

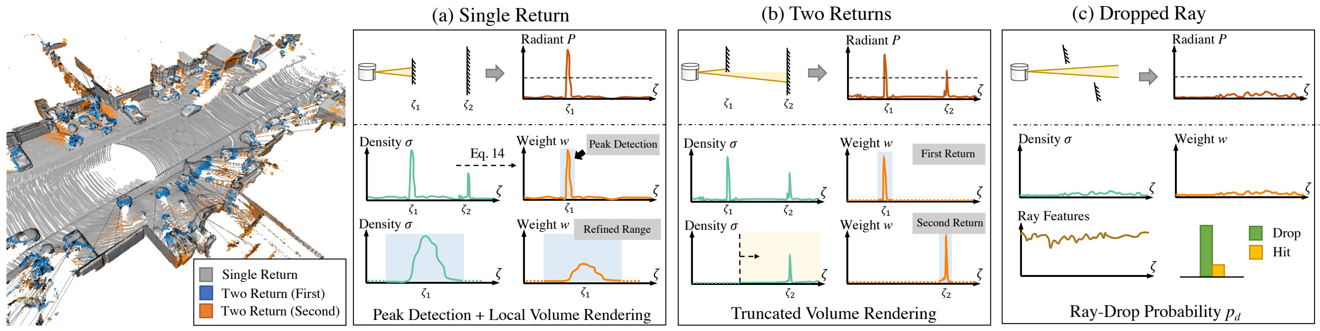


Figure 1. Left: real LiDAR scan demonstrating key LiDAR return properties: a single return and two returns (first return shown in blue and second return in orange). Right: NFL models the waveform and accurately reproduces these properties. (a) Top: the LiDAR energy is fully scattered by the first surface. Bottom: NFL estimates range via peak detection on the computed weights w followed by volume rendering based range refinement. (b) Top: secondary returns resulting from a beam hitting two surfaces. Bottom: NFL employs beam divergence and a truncated volume rendering to estimate the second return. (c) Top: beams that do not hit a surface do not return detectable signal. Bottom: NFL utilizes geometric and semantic features to predict the ray drop probability. Refer to section 4.3 for more details.

σ_ζ at range ζ is a scalar function that indicates the differential likelihood of hitting a reflective particle at position $\mathbf{r}_\zeta = \mathbf{o} + \zeta\mathbf{d}$. *Transmittance* T_ζ indicates the probability of traversing the interval $[0, \zeta)$ without hitting anything. Taking a differential step $d\zeta$ along the ray, the probability of *not* hitting anything is $T_{\zeta+d\zeta} = T_\zeta \cdot (1 - \sigma_\zeta d\zeta)$. Integrating over an interval $[\zeta_0, \zeta)$ yields the probability $T_{\zeta_0 \to \zeta}$ of traversing the interval unhindered,

$$T_{\zeta_0 \to \zeta} \equiv \frac{T_\zeta}{T_{\zeta_0}} = \exp\left(-\int_{\zeta_0}^{\zeta} \sigma_t dt\right) , \tag{1}$$

leading to the decomposition: $T_\zeta = T_{0 \to \zeta_0} \cdot T_{\zeta_0 \to \zeta}$.

Integration over homogeneous media. Assuming a homogeneous medium along the ray segment $[\zeta_j, \zeta_{j+1}]$ with constant radiance $\mathbf{c} \in \mathbb{R}^3$ and density σ, the accumulated radiance from that segment evaluates to

$$\mathbf{c}(\zeta_j \to \zeta_{j+1}) = \mathbf{c}_{\zeta_j} \int_{\zeta_j}^{\zeta_{j+1}} T_{\zeta_j \to \zeta} \cdot \sigma_\zeta \, d\zeta = \alpha_{\zeta_j} \mathbf{c}_{\zeta_j} \ , \tag{2}$$

with $\alpha_{\zeta_j} = 1 - \exp\big(-\sigma_{\zeta_j}(\zeta_{j+1} - \zeta_j)\big)$ being the *opacity*.

Volume rendering. By discretizing the ray into N segments with piecewise constant densities and radiance values, we obtain the total irradiance (color to be rendered):

$$\mathbf{c} = \sum_{j=1}^{N} \int_{\zeta_j}^{\zeta_{j+1}} T_\zeta \cdot \sigma_\zeta \mathbf{c}_\zeta \, d\zeta = \sum_{j=1}^{N} w_j \mathbf{c}_{\zeta_j} \ , \tag{3}$$

where w_j is the *weight* for the j-th segment:

$$w_j = \alpha_{\zeta_j} \prod_{k=1}^{j-1} (1 - \alpha_{\zeta_k}) \ . \tag{4}$$

3.2. LiDAR model

LiDAR emits laser beam pulses and determines the distance from the sensor to the nearest reflective surface by measuring the time of flight. Often the LiDAR beams are pictured as ideal straight-line segments ending on a 3D surface point. In reality, things are more complicated: real lasers emit a pulse with non-zero divergence and finite pulse width, while real receivers employ signal processing techniques like radiant thresholding and binning to detect the return. This leads to phenomena such as discretization errors, over- and underestimation biases (*cf*. Fig. 2), and multiple returns from one beam (or no return at all). In the following, we discuss key aspects of the LiDAR acquisition process and explain the effects that emerge, which inspire our model design. We also built a LiDAR simulator that accounts for these mechanisms, see Sec. 5.1.

Beam divergence. LiDAR beams diverge as they travel away from the sensor. The size of laser beams can become wider over distance, and typically not negligible in street scenes. Consequently, the illuminated area grows and the irradiance (radiant power per area) decreases with increasing range. The size of the beam's footprint is characterised by the divergence angle $(2\gamma_0)$ and the range ζ. Let $\mathbf{r}^\gamma$ be an ideal ray within the beam's cross-section, $\gamma \leq \gamma_0$, then its irradiance $E(\zeta, \gamma)$ at range ζ can be approximated by a Gaussian function in the ray coordinate system [47]:

$$E(\zeta, \gamma) = \frac{2I_0}{\pi(\gamma_0\zeta)^2} g(\gamma), \quad g(\gamma) = \exp\left(-2\frac{\gamma^2}{\gamma_0^2}\right), \tag{5}$$

where I_0 is the pulse peak power.

Pulse waveform. When the emitted LiDAR pulse returns to the sensor, the range to the reflective surface can be determined from its travel time and the speed of light c. Since the pulse has finite duration τ_H, the time of return is found by analysing the received intensity profile. The transmitted pulse power over time can be characterised as [6]:

$$P_e(t) \propto \left(\frac{t}{\tau}\right)^2 \exp\left(-\frac{t}{\tau}\right), \quad \tau = \frac{\tau_H}{1.75} . \tag{6}$$

The range-dependent received radiant power $P(\zeta)$ is the result of convolving the pulse power with the systems impulse response $H(\zeta)$ [12, 13, 37]:

$$P(\zeta) = \int_0^{2\zeta/c} P_e(t) H(\zeta - \frac{ct}{2})\, dt , \tag{7}$$

where the impulse response $H(\zeta)$ is a composition of the target and the receiver responses: $H(\zeta) = H_T(\zeta) H_C(\zeta)$. Assuming a Lambertian surface, the target response due to a surface located at range ζ_0 depends on the incidence angle θ and the reflectance ρ:

$$H_T(\zeta) = \frac{\rho}{\pi} \cos(\theta) \delta(\zeta - \zeta_0) , \tag{8}$$

with $\delta(\cdot)$ the Dirac delta function. The receiver response $H_C(\zeta)$ is computed by integrating over the solid angle spanned by the receiver's effective area A_e:

$$H_C(\zeta) = T_\zeta^2 \frac{A_e}{\zeta^2} , \tag{9}$$

where $T_\zeta \in [0, 1]$ is the one-way transmittance, squared to account for the two-way trip.

Beam discretization. In practice, we follow [49] and approximate the Gaussian beam profile using $M = 37$ rays that are radially distributed around the central ray with different divergence angles γ_i. The total radiant power $P(\zeta)$ is the weighted sum over those rays: $P(\zeta) = \sum_{i=1}^{M} g(\gamma_i) P_i(\zeta)$. Taking into account the beam divergence is important to reproduce two important phenomena: range biases and multiple returns, see Fig. 1 and Fig. 2. As different rays hit a slanted surface at different ranges the integrated waveform peak may shift, causing over- or underestimations. Along object edges, rays within the same beam may hit different surfaces, causing multiple peaks (respectively, range readings), in the return waveform.

Range estimation. One common approach to estimate the surface range from the received waveform is to locate its peak. To that end the signal is discretized in time to obtain a histogram, and local maxima above a certain threshold are declared detections [49]. The associated range values are then corrected to remove known biases stemming from the pulse waveform (*cf*. Eq. (6)) and, optionally, biases due to the radiant power [49]. By modeling the binning and thresholding procedure one can reproduce further LiDAR behaviors: systematic discretization errors in the range resolution (*cf*. Fig. 2), and the dropping of rays with low returned power (*cf*. Fig. 1).

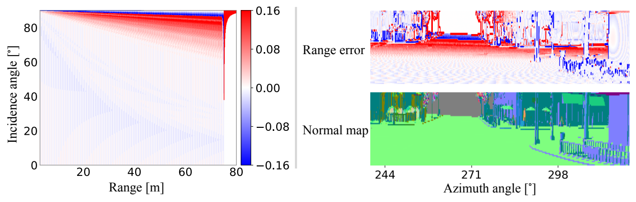


Figure 2. The range accuracy of the LiDAR sensor is affected by waveform discretization and beam divergence. The LiDAR sensor has a tendency to overestimate range in high incidence angle regime, which becomes increasingly pronounced at higher range regimes (left). This is also reflected on *TownReal* dataset (right).

4. LiDAR Novel View Synthesis

We now turn to constructing a neural field model tailored for LiDAR scans, along with a differentiable volume rendering scheme to enable LiDAR novel view synthesis. We first formulate the problem setting, then set up a corresponding neural scene representation (Sec. 4.1) and derive volume rendering for active sensing (Sec. 4.2). Finally, we describe the rendering procedure used to synthesize novel views (Sec. 4.3) and our optimisation scheme (Sec. 4.4).

Problem setting. Consider a collection of LiDAR scans $\mathcal{X} = \{\mathbf{X}_v\}_{v=1}^{n_v}$ captured by a moving sensor (*e.g*., mounted on a vehicle). Each scan $\mathbf{X}_v$ is associated with a sensor pose $\mathbf{T}_v \in \mathrm{SE}(3)$ and consists of n_r rays. Every ray $\mathbf{r}(\mathbf{o}, \mathbf{d})$ records observations $(\zeta_1, e_1, p_d, p_s, \zeta_2, e_2)$: the range ζ_1 and intensity e_1 of the first return; a ray drop flag $p_d \in \{0, 1\}$; a two-return mask $p_s \in \{0, 1\}$; and range ζ_2 and intensity e_2 values of the second return. Our goal is to reconstruct a (continuous) volumetric representation of the scene in terms of density σ and reflectance ρ, from which we can subsequently render virtual LiDAR scans $\mathbf{X}_{tgt}$ from novel sensor poses $\mathbf{T}_{tgt}$.

4.1. Neural scene representation

We encode the scene as a neural field $F : (\mathbf{x}, \mathbf{d}) \mapsto (\sigma, \rho, p_d)$ that takes as input a location $\mathbf{x} \in \mathbb{R}^3$ and viewing direction $\mathbf{d} \in \mathbb{R}^3$, and returns a density σ, a reflectance ρ and a ray drop probability p_d. We found it beneficial to additionally return also a local contribution $p_d \in [0, 1]$ to the probability of ray drop, which will be discussed below. Technically, we use a hash encoding [30] to map coordinates $\mathbf{x}$ to positional features $\mathbf{f}_{\text{pos}} \in \mathbb{R}^{32}$ and project the view direction onto the first 16 coefficients of the spherical harmonics basis, $\mathbf{f}_{\text{dir}} \in \mathbb{R}^{16}$. The neural field is parameterized by four Multi-Layer Perceptrons (MLPs): $[\sigma; \mathbf{f}_{\text{geo}}] = f_\sigma(\mathbf{f}_{\text{pos}})$ regresses density and extracts an additional geometry feature $\mathbf{f}_{\text{geo}} \in \mathbb{R}^{15}$ that supports the other networks; $\rho = f_\rho(\mathbf{f}_{\text{geo}}, \mathbf{f}_{\text{dir}})$ regresses reflectance; $p_d = f_{\text{drop}}(\mathbf{f}_{\text{geo}}, \mathbf{f}_{\text{dir}})$ classifies whether a ray drop occurs; and $p_s = f_{\text{sr}}(\mathbf{f}_{\text{beam}})$ classifies the existance of a second return. The feature $\mathbf{f}_{\text{beam}}$

will be detailed in Sec. 4.3.

4.2. Volume rendering for LiDAR rays

In contrast to passive sensors like cameras that rely on ambient illumination, LiDAR actively illuminates the scene and measures the back-scattered radiance. This two-way transmittance alters the volume rendering formulation.

Radiant power integration. As discussed in Sec. 3.2 the radiant power along a LiDAR ray is a delta function that is non-zero only at reflecting surfaces. To incorporate this forward model into the volumetric representation we combine Eq. (8) and Eq. (9) to obtain the probabilistic radiant power:

$$P_\zeta = C\frac{T_\zeta^2 \cdot \sigma_\zeta \rho_\zeta}{\zeta^2}\cos(\theta)\ , \tag{10}$$

where C is a system constant, ρ_ζ is the differentiable reflectance, and θ is the incidence angle. In a homogeneous medium with constant reflectance ρ and density σ, the integrated $P(\zeta_j \rightarrow \zeta_{j+1})$ evaluates to:

$$P(\zeta_j \rightarrow \zeta_{j+1}) = \int_{\zeta_j}^{\zeta_{j+1}} C\frac{T^2_{\zeta_j \rightarrow \zeta}\sigma_\zeta \rho_\zeta}{\zeta^2}\cos(\theta_j)\, d\zeta \approx \alpha_{\zeta_j}\rho'_{\zeta_j}, \tag{11}$$

where we approximate $\zeta \in [\zeta_j, \zeta_{j+1}]$ by $\frac{\zeta_j + \zeta_{j+1}}{2}$, and

$$\alpha_{\zeta_j} = \frac{1}{2}\left(1 - e^{-2\sigma_{\zeta_j}\delta_j}\right)\ ,\ \rho'_{\zeta_j} = C\rho_{\zeta_j}\frac{4\cos(\theta_j)}{(\zeta_j + \zeta_{j+1})^2}. \tag{12}$$

Volume rendering. The observed power at the active sensor can be evaluated by plugging Eq. (11) into Eq. (3):

$$P = \sum_{j=1}^{N}\int_{\zeta_j}^{\zeta_{j+1}} C\frac{T_\zeta^2 \cdot \sigma_\zeta \rho_\zeta}{\zeta^2}\cos(\theta_j)\, d\zeta = \sum_{j=1}^{N} w_j \rho'_{\zeta_j}, \tag{13}$$

where the weights w_j are now evaluated as (*cf*. Eq. (4)):

$$w_j = 2\alpha_{\zeta_j} \cdot \prod_{k=1}^{j-1}(1 - 2\alpha_{\zeta_k})\ . \tag{14}$$

4.3. Assembling the beam from multiple rays

Next, we apply the adapted volume rendering formulation to multiple rays within a single LiDAR beam.

First range estimation. We adopt a two-stage approach to extract range values from the neural field,[2] as shown in Fig. 1 (a). To estimate the range for an ideal ray $\mathbf{r}$, we uniformly sample N^c points and query their density values, then compute the weights $\{w_j^c\}_{j=1}^{N^c}$ using Eq. (14). A coarse peak estimate ζ_p is obtained by finding the point with the highest weight along the ray: $p = \arg\max_j \{w_j^c\}_{j=1}^{N^c}$. Next, we uniformly sample N^f points from the local interval $\zeta_j \in [\zeta_p - \epsilon, \zeta_p + \epsilon]$. The weights w_j^f at these points are recomputed and normalized to then obtain the final, refined range estimate ζ_f as: $\zeta_f = \sum_{j=1}^{N^f} w_j^f \cdot \zeta_j$.

[2]Note the similarity to the detector in the instrument that first finds the peak of the waveform, then corrects for pulse shape.

Second range estimation. As discussed in Sec. 3.2 a single LiDAR beam might have multiple returns if enough energy was reflected from surfaces further away than the first return. To capture this behavior in our scene representation, we employ *truncated* volume rendering to estimate the radiant power beyond the first return (see Fig. 1 (b)).

Specifically, for each beam, we first predict a two-return mask p_s, by classifying its features $\mathbf{f}_{\text{beam}} = (\bar{\mathbf{f}}_{\text{geo}}, \mathbf{f}_{\text{dir}}, \mathbf{f}_{\text{range}})$, where $\bar{\mathbf{f}}_{\text{geo}}$ is the volume-rendered geometric feature, and $\mathbf{f}_{\text{range}}$ describes the standard deviation and maximum discrepancy of range estimates at the first return. Intuitively, $\bar{\mathbf{f}}_{\text{geo}}$ describes the local geometry (*e.g*. an edge), $\mathbf{f}_{\text{dir}}$ encodes the relation of the beam to the geometry, and $\mathbf{f}_{\text{range}}$ characterizes the beam's prior interaction with the scene.

For beams that have two returns, we then perform *truncated* volume rendering as follows. We first add a buffer ξ[3] to the estimated range ζ_1 of the first return. We then reset the transmittance $T_{\zeta_1+\xi}$ to 1 by zeroing out the densities up to $(\zeta_1 + \xi)$ and recalculate the weights to ensure that they adhere to Eq. (14). Finally, we repeat the range estimation described above to estimate the range of the second return ζ_2. Note that for beams with two returns, the estimated range ζ_1 denotes the minimum range of all rays within the beam diameter, i.e. we perform volume rendering on all rays of a beam and pick the closest one as the first return. This is different from the beams with a single return where we directly use the central ray to estimate ζ_1.

Reflectance estimation. At every detected surface point we can also retrieve reflectance from the neural field, using the relation $\rho = \sum_{j=1}^{N^f} w_j^f \cdot \rho_j$.

Ray drop probability. In real LiDAR sensors, some emitted beams return no range measurement at all. This happens when the observed return signal has either too low amplitude or no clear peak (Fig. 1 (c)). However, this effect is hard to model in a fully physics-based way,[4] because it depends on (usually undisclosed) details of the detection logic. We empirically observe that the ray drop probability can be learned from LiDAR measurements. To this end, we augment the neural scene representation with a dedicated variable for the local probability of *not* back-scattering radiant power $p_d(\zeta) \in \{0, 1\}$[5]. Volume ren-

[3]The buffer ξ is sensor specific and describes the minimum spacing between two distinct returns.

[4]Beyond simple thresholding, which our beam model would support.

[5]Please refer to the supplementary for discussions on this design choice.

dering integrates that quantity into a ray drop probability: $p_d(\mathbf{r}) = \sum_{j=1}^{N_c} w_j^c \cdot p_d(\zeta_j)$.

4.4. Training the neural LiDAR field

Given a set of posed LiDAR scans, we optimise our neural field model by minimising the loss

$$\mathcal{L} = \mathcal{L}_{\text{range}} + \lambda_e \mathcal{L}_e + \lambda_d \mathcal{L}_d + \lambda_s \mathcal{L}_s \ , \tag{15}$$

consisting of a reconstruction loss $\mathcal{L}_{\text{range}}$ for range estimation, reflectance loss $\mathcal{L}_e$, and classification losses $\mathcal{L}_d$ for ray drops and $\mathcal{L}_s$ for two returns.

Range reconstruction. We add two separate losses for the coarse range ζ_p and the refined range ζ_f, $\mathcal{L}_{\text{range}} = \mathcal{L}^c_{\text{range}} + \mathcal{L}^f_{\text{range}}$. For coarse range, we impose a Gaussian distribution [38] around the ground truth $\hat{\zeta}$,

$$\mathcal{L}^c_{\text{range}} = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \left(1 - \sum_{w_j \in \mathcal{X}^n_c} w_j \hat{w}_j + \sum_{w_k \in \mathcal{X}^e_c} w_k^2 \right) \ , \tag{16}$$

where $\mathcal{R}$ is the set of LiDAR rays, $\mathcal{X}^n_c$ and $\mathcal{X}^e_c$ denote points sampled within and outside the interval $[\hat{\zeta} - \epsilon, \hat{\zeta} + \epsilon]$. The ground truth weight $\hat{w}_j$ is calculated by integrating the Gaussian distribution. The range refinement loss is defined as: $\mathcal{L}^f_{\text{range}} = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} |\hat{\zeta} - \zeta_f|$.

Reflectance reconstruction is optimized by minimizing an L2 loss w.r.t. the ground truth intensity $\hat{e}$: $\mathcal{L}_e = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} (\hat{e} - e)^2$.

Ray drop and dual return masks are trained as classification tasks, by minimizing the combination of a binary cross entropy loss $\mathcal{L}_{bce}$ and a Lovasz loss $\mathcal{L}_{ls}$ [4]:

$$\mathcal{L}_* = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \left(\mathcal{L}_{bce}(p_*, \hat{p_*}) + \mathcal{L}_{ls}(p_*, \hat{p_*}) \right) \ . \tag{17}$$

5. Experiments

We start by describing our LiDAR simulator, datasets, evaluation metrics, and baselines in Sec. 5.1. In Sec. 5.2, we evaluate NFL directly on the LiDAR novel view synthesis task. Finally, in Sec. 5.3 we evaluate the suitability of our synthesized LiDAR data for two low-level tasks, point cloud registration and semantic segmentation.

5.1. Datasets and Evaluation setting

LiDAR simulator – TownReal dataset. To enable quantitative evaluation in a controlled environment, we build a LiDAR simulator that allows us to virtually scan synthetic 3D assets represented either as triangular meshes or surfels. Specifically, we follow the LiDAR model described in Sec. 3.2 and allow control over the angular resolution, beam divergence, and pulse shape of the LiDAR sensor.

We use this simulator in combination with a 3D asset of a town [16] to synthesize the *Town* dataset. We generate four scenes by splitting the 3D asset into four non-overlapping areas. Training and test scans are created from different trajectories. We use two different configurations of the LiDAR sensor: *(i) TownClean*, in which LiDAR scans are simulated using an idealized, non-divergent ray; and *(ii) TownReal*, with a diverged beam profile approximated via 37 subrays. See the supplementary material for further details.

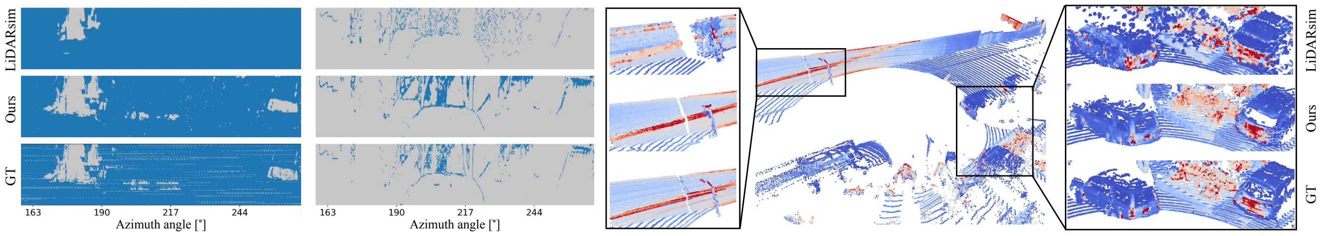
Waymo Open dataset. For evaluation on real-world data we use Waymo open dataset [43] which was captured by a 64-beam LiDAR sensor at 10 Hz. Here, we select four static scenes (see sequence IDs in supplementary material) and extract a five-second clip from each, resulting in 50 scans per scene. We hold out every 5-th frame as a test view and use the remaining 40 scans for training (*Waymo Interp.*)

To evaluate the methods in a more challenging setting we propose a novel evaluation protocol based on a closed-loop simulation (*Waymo NVS*). The protocol involves training and testing on all scans of a scene by first optimizing on the input views to synthesize novel views from a changed trajectory (shift the sensor by [1.5, 1.5, 0.5] meters[6]). The novel view are then used to re-optimize the method, synthesize scans in the original view and compare to the original scans to gauge performance. This formulation allows us to control task difficulty and could also be applied to evaluate camera-based novel view synthesis methods.

Evaluation metrics. To evaluate range accuracy, we report four metrics: mean and median absolute errors (*MAE* [cm], *MedAE* [cm]), two-way Chamfer distance (*CD* [cm]), and *recall@50*, which denotes the percentage of rays with range errors below 50 cm. We additionally measure the two return segmentation recall (*Seg. recall*) and precision (*Seg. precision*). Intensity is evaluated using mean absolute error (*MAE*). For ray drop segmentation, we report recall and precision [%], and intersection-over-union (*IoU* [%]). For point cloud registration, we report rotation error (*RE* [°]) and translation error (*TE* [cm]).

Baselines. We compare NFL to four baselines. Closest to our problem setup is LiDARsim [24] which was designed for LiDAR synthesis based on surface reconstruction and ray-surfel casting. We re-implement LiDARsim and augment it with a diverged beam profile to enable synthesis of the second returns. Additionally, we adapt three NeRF-like methods that were originally proposed for image synthesis i-NGP [30], DS-NeRF [8], and URF [38] by modifying their volumetric rendering to improve their range predictions. Additional details are available in the supplementary.

[6]Please refer to the supplementary for ablations on different sensor shift configurations.



(a) First return mask (b) Second return mask (c) LiDAR scans coloured by intensity values

Figure 3. Qualitative results of LiDAR novel view synthesis on *Waymo Interp.* dataset. On the left, we color-code rays with and without return. On the right side, LiDAR intensity values are color-coded as :0 ▬ 0.25.

Method		First range			Second range					Intensity		Ray drop		
		Recall@50↑	MAE↓	MedAE↓	Seg. recall↑	Seg. precision↑	Recall@50↑	MAE↓	MedAE↓	$\text{MAE}^{1\text{st}}$↓	$\text{MAE}^{2\text{nd}}$↓	Recall↑	Precision↑	IoU↑
LiDARsim [24]		74.1	105.4	18.5	3.5	11.5	1.0	2258.0	1898.2	0.013	0.018	32.5	**85.5**	30.5
Ours	Central ray	92.8	32.8	5.6	79.8	62.9	61.1	589.1	21.8	0.004	0.009	64.3	81.7	**57.1**
	Diverged beam	92.3	36.1	5.7	82.1	55.6	67.4	505.1	13.4	**0.004**	**0.008**	**65.1**	78.0	56.1
	GT mask	**93.2**	**29.7**	**5.6**	**100.0**	**100.0**	**79.8**	**116.0**	**8.1**	0.004	0.011	65.1	78.0	56.1

Table 1. Comprehensive ray measurement evaluation of LiDAR novel view synthesis on *Waymo Interp.* dataset.

	TownClean			TownReal			Waymo interp.			Waymo NVS		
Method	MAE↓	MedAE↓	CD↓	MAE↓	MedAE↓	CD↓	MAE↓	MedAE↓	CD↓	MAE↓	MedAE↓	CD↓
i-NGP [30]	42.2	4.1	17.4	49.8	4.8	19.9	**26.4**	5.5	**11.6**	**30.4**	7.3	15.3
DS-NeRF [8]	41.7	3.9	16.6	48.9	4.4	18.8	28.2	6.3	14.5	30.4	7.2	16.8
URF [38]	43.3	4.2	16.8	52.1	5.1	20.7	28.2	5.4	12.9	43.1	10.0	21.2
LiDARsim [24]	159.6	**0.8**	23.5	162.8	3.8	27.4	116.3	15.2	27.6	160.2	16.2	34.7
Ours	**32.0**	2.3	**9.0**	**39.2**	**3.0**	**11.5**	30.8	**5.1**	12.1	32.6	**5.5**	**13.2**

Table 2. Results of LiDAR novel view synthesis for the first range.

	TownClean			Waymo Interp.		
Method	MAE↓	MedAE↓	CD↓	MAE↓	MedAE↓	CD↓
i-NGP [30]	41.0 (-1.2)	4.1 (0.0)	17.6 (0.2)	25.3 (-1.1)	4.5 (-1.0)	10.5 (-1.1)
DS-NeRF [8]	37.4 (-4.2)	3.0 (-0.9)	14.4 (-2.2)	27.4 (-0.8)	5.4 (-1.0)	13.6 (-0.9)
URF [38]	46.4 (3.0)	4.5 (0.3)	18.4 (1.6)	28.3 (0.1)	5.3 (-0.1)	13.1 (0.2)
Ours	32.0 (-2.1)	2.3 (-2.5)	9.0 (-3.9)	30.8 (-2.1)	5.1 (-2.0)	12.1 (-2.3)

Table 3. Ablation study of volume rendering for active sensing.

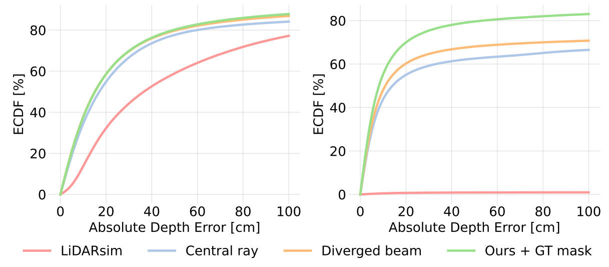


Figure 4. Beam divergence modeling improves range accuracy of rays with dual returns. This is evident in the improved error distribution of the first (left) and second return range (right).

5.2. Evaluation of LiDAR novel view synthesis

Ray measurement. Using the *Waymo Interp.* dataset we conduct a comprehensive analysis of all the ray measurements and present the results in Tab. 1 and Fig. 3. Only NFL and LiDARsim [24] are used for this experiment, as other baselines can only support a single return. LiDARsim's surface representation is explicit and not optimized for novel view synthesis nor accounts for view-dependent effects, which results in inferior range prediction and difficulties in retrieving secondary returns. In contrast, NFL directly optimizes the neural field for view synthesis while accounting for LiDAR acquisition process characteristics, resulting in significantly reduced range errors and superior performance in intensity and ray drop probability estimation. Notably, equipping our model with a *diverged beam* representation improves range estimation for both first and second returns for rays with dual returns(*cf*. Fig. 4). However, diverged beam does slightly degrade the overall first-range accuracy likely due to imprecise two-return mask estimation. This hypothesis is supported by results using the ground truth two-return mask (*GT mask*). In Fig. 5 we show more qualitative results of novel view synthesis by NFL.

First range. The results of estimating the range of the first return on all datasets are presented in Tab. 2 and Fig. 6. As demonstrated by the results on *TownClean*, *TownReal*, and *Waymo NVS*, the proposed volume rendering formulation of NFL effectively regularizes the density field resulting in superior performance in challenging cases. Even in the easier setting (resembling overfitting) on *Waymo Interp.* dataset, our method achieves competitive performance. In contrast, NeRF-like formulations (i-NGP [30], DS-NeRF [8], and URF [38]) perform poorly when evaluated on real novel views due to their inability to account for the active sensing principle. LiDARsim achieves promising results on datasets with simple geometry and clean LiDAR measurements, as evidenced by low *MedAE* scores on *TownClean*

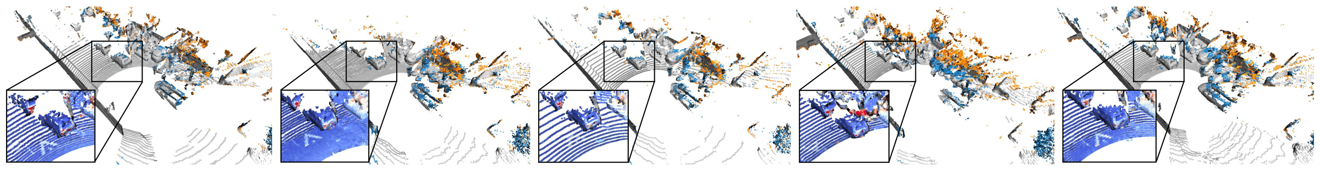

Ref. frame (θ_0, x_0, y_0, z_0)
(a) $\theta = \theta_0 - 5$
(b) 64 →32 beam; $\theta = \theta_0 - 2$
(c) $z = z_0 + 1$
(d) $x = x_0 + 1$; $y = y_0 + 1$

Figure 5. LiDAR novel view synthesis by changing the sensor elevation angle $\theta[°]$, pose $(x, y, z)[m]$ and number of beams. Zoom-in points are color-coded by intensity values.

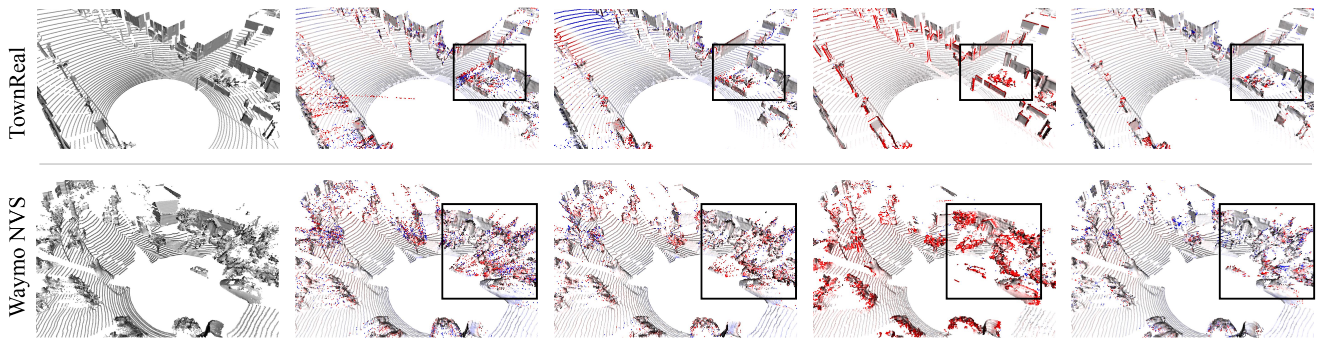


Figure 6. Qualitative comparison of first range estimation. Regions with gross errors (-100 100 cm) are highlighted.

Method	TownClean			TownReal			Waymo NVS		
	Rec@5 ↑	RE ↓	TE ↓	Rec@5 ↑	RE ↓	TE ↓	Rec@2 ↑	RE ↓	TE ↓
i-NGP [30]	70.3	0.1	4.2	76.0	0.1	4.2	60.2	0.1	1.9
DS-NeRF [8]	58.3	0.2	5.1	56.2	0.2	5.1	42.3	0.1	2.4
URF [38]	61.5	0.2	5.0	59.9	0.1	4.7	32.1	0.1	2.7
LiDARsim [24]	**82.8**	0.1	**3.4**	<u>79.2</u>	0.1	**3.4**	<u>62.8</u>	0.1	<u>1.8</u>
Ours	<u>80.2</u>	0.1	<u>3.7</u>	**85.9**	0.1	**3.4**	**71.9**	0.1	**1.7**

Table 4. Point cloud registration results on three datasets.

Method	Vehicle			Background		
	Recall ↑	Precision ↑	IoU ↑	Recall ↑	Precision ↑	IoU ↑
i-NGP [30]	<u>93.2</u>	85.9	<u>80.9</u>	98.3	<u>99.2</u>	<u>97.6</u>
DS-NeRF [8]	90.7	**87.1**	80.2	**98.5**	98.9	97.4
URF [38]	87.8	81.7	73.7	98.0	98.4	96.5
Lidarsim [24]	90.5	70.5	65.9	94.9	99.0	94.0
Ours	**95.9**	<u>87.0</u>	**83.9**	<u>98.3</u>	**99.5**	**97.8**

Table 5. Semantic segmentation results on *Waymo NVS* dataset.

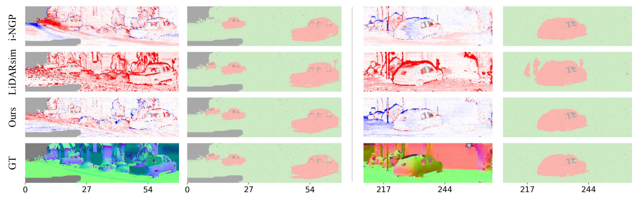


Figure 7. Semantic segmentation results on synthesised *Waymo NVS* dataset. Geometry in-accuracy (-100 100 cm) leads to erroneous semantic segmentation (dropped rays, vehicle, pedestrian, background).

and *TownReal*. However, its explicit representation struggles with complex geometry in noisy real-world scenes, *e.g.*, the vegetation regions in the *Waymo* dataset, resulting in high *MedAE* scores.

Ablation study of volume rendering for active sensing. To evaluate the effectiveness of our volume rendering formulation for active sensors, we replace the volume rendering [29] formulation initially developed for passive sensing in all NeRF-based baselines and report performance difference in Tab. 3. Our formulation improves range accuracy across all settings, without any hyper-parameter tuning.

5.3. Downstream evaluation of novel views

Having demonstrated NFL's improved ability to synthesize high-quality LiDAR scans through various metrics, we proceed to evaluate their perceptual quality by using them as input for two low-level perception tasks: point cloud registration [15] and semantic segmentation [45].

Point cloud registration. To evaluate the extent to which synthesized scans preserve local geometric features, we apply the same point cloud registration model [14] pre-trained on Waymo [43] to both GT LiDAR scans and scans synthesized using different methods. Tab. 4 shows that NFL outperforms the baseline methods on datasets with complex geometry and higher noise levels (*TownReal* and *Waymo NVS*) that are more susceptible to artifacts occurring as a result of the LiDAR acquisition process.

Semantic segmentation. To probe the potential domain gap between real and synthetic scans we apply the same, pre-trained semantic segmentation model [45] to both and compare the predictions. Tab. 5 depicts the performance for both the *vehicle* and *background* classes. Notably, NFL achieves the highest recall for the *vehicle* class, which is strongly affected by dual returns and ray drops. Example predictions are shown in Fig. 7.

6. Limitations and future work

We have presented NFL, a neural field-based approach for synthesizing LiDAR scans from novel viewpoints. NFL combines the benefits of volume rendering with a physically based model of LiDAR acquisition process to faithfully model LiDAR characteristics including beam divergence, secondary returns, and ray dropping. Even though NFL significantly outperforms explicit reconstruct-then-simulate methods as well as other NeRF-style methods, it still has some limitations that we would like to address in future work. Firstly, with NFL, we try to seek a balance between adhering to the physical principles of LiDAR and incorporating semantic features and learning. While our formulation already shows improved performance over baselines, there is still potential for further improvements. For example, while real-world LiDAR sensors perform range detection on the integrated beam radiant, we actually found that using the density-based weights separately for each ray leads to improved performance. Prediction of the second return mask enables us to model secondary returns and to further improve the estimation of the first return. Yet, as indicated by the oracle study, improving the mask prediction could lead to further improvements. Finally, our method is based on a NeRF-style representation and therefore requires per-scene optimization. Generalization across scenes and handling dynamic environments are key challenges that we plan to address in future work.

Acknowledgements. We sincerely thank Benjamin Naujoks, Steven Butrimas, Tomislav Medić, Yu Han, and Prof. Dr. Andreas Wieser for helpful discussions around LiDAR models. We are grateful for the feedback on figures from Rodrigo Caye Daudt. This appreciation extends to Zvi Greenstein for organisation support.

References

[1] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. TöRF: Time-of-flight radiance fields for dynamic scene view synthesis. *NeurIPS*, 2021.

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *CVPR*, 2021.

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.

[4] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The Lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018.

[5] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *CVPR*, 2020.

[6] Tomas Carlsson, Ove Steinvall, and Dietmar Letalick. Signature simulation and signal analysis for 3-d laser radar. Technical report, Swedish Defence Research Agency, 2001.

[7] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. GARF: Gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *arXiv preprint arXiv:2204.05735*, 2022.

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. *arXiv preprint arXiv:2107.02791*, 2021.

[9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *CoRL*, 2017.

[10] Jin Fang, Dingfu Zhou, Feilong Yan, Tongtong Zhao, Feihu Zhang, Yu Ma, Liang Wang, and Ruigang Yang. Augmented LiDAR simulator for autonomous driving. *IEEE RA-L*, 5(2):1931–1938, 2020.

[11] Benoit Guillard, Sai Vemprala, Jayesh K Gupta, Ondrej Miksik, Vibhav Vineet, Pascal Fua, and Ashish Kapoor. Learning to simulate realistic LiDARs. *arXiv preprint arXiv:2209.10986*, 2022.

[12] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. LiDAR snowfall simulation for robust 3d object detection. *arXiv preprint arXiv:2203.15118*, 2022.

[13] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real LiDAR point clouds for 3d object detection in adverse weather. In *CVPR*, 2021.

[14] Shengyu Huang, Zan Gojcic, Jiahui Huang, Andreas Wieser, and Konrad Schindler. Dynamic 3d scene analysis by point cloud accumulation. In *ECCV*, 2022.

[15] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, 2021.

[16] Kasiopy. Town with suburb. https://www.turbosquid.com/3d-models/town-suburb-3d-max/1085661. last accessed 2023.

[17] Velat Kilic, Deepti Hegde, Vishwanath Sindagi, A Brinton Cooper, Mark A Foster, and Vishal M Patel. Lidar light scattering augmentation (LISA): Physics-based simulation of adverse weather conditions for 3d object detection. *arXiv preprint arXiv:2107.07004*, 2021.

[18] Simon Klenk, Lukas Koestler, Davide Scaramuzza, and Daniel Cremers. E-NeRF: Neural radiance fields from a moving event camera. *arXiv preprint arXiv:2208.11300*, 2022.

[19] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IROS*, 2004.

[20] Akhil Kurup and Jeremy Bos. DSOR: A scalable statistical filter for removing falling snow from LiDAR point clouds in severe winter weather. *arXiv preprint arXiv:2109.07078*, 2021.

[21] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. In *ICCV*, 2021.

[22] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020.

[23] Andrew Luo, Yilun Du, Michael J Tarr, Joshua B Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *arXiv preprint arXiv:2204.00628*, 2022.

[24] 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*, 2020.

[25] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[26] Nelson Max and Min Chen. Local and global illumination in the volume rendering integral. Technical report, Lawrence Livermore National Lab., Livermore, CA, 2005.

[27] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019.

[28] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. In *CVPR*, 2022.

[29] 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. In *ECCV*, 2020.

[30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022.

[31] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *CVPR*, 2021.

[32] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *CVPR*, 2021.

[33] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.

[34] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Computer Graphics and Interactive Techniques*, 2000.

[35] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *CVPR*, 2021.

[36] Mohamad Qadri, Michael Kaess, and Ioannis Gkioulekas. Neural implicit surface reconstruction using imaging sonar. *arXiv preprint arXiv:2209.08221*, 2022.

[37] Ralph H Rasshofer, Martin Spies, and Hans Spies. Influences of weather phenomena on automotive laser radar systems. *Advances in Radio Science*, 9:49–60, 2011.

[38] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. *arXiv preprint arXiv:2111.14643*, 2021.

[39] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. *arXiv preprint arXiv:2112.03288*, 2021.

[40] Viktor Rudnev, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. EventNeRF: Neural radiance fields from a single colour event camera. *arXiv preprint arXiv:2206.11896*, 2022.

[41] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[42] Yi-Chien Shih, Wei-Hsiang Liao, Wen-Chieh Lin, Sai-Keung Wong, and Chieh-Chih Wang. Reconstruction and synthesis of lidar point clouds of spray. *IEEE RA-L*, 7(2):3765–3772, 2022.

[43] 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*, 2020.

[44] Andrea Tagliasacchi and Ben Mildenhall. Volume rendering digest for nerf. *arXiv preprint arXiv:2209.02417*, 2022.

[45] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, 2020.

[46] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *ECCV*, 2014.

[47] Wolfgang Wagner, Andreas Ullrich, Vesna Ducic, Thomas Melzer, and Nick Studnicka. Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(2):100–112, 2006.

[48] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[49] Lukas Winiwarter, Alberto Manuel Esmorís Pena, Hannah Weiser, Katharina Anders, Jorge Martínez Sánchez, Mark Searle, and Bernhard Höfle. Virtual laser scanning with HELIOS++: A novel take on ray tracing-based simulation of topographic full-waveform 3d laser scanning. *Remote Sensing of Environment*, 269:112772, 2022.

[50] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022.

[51] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 2021.

[52] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021.

[53] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.