Supplementary Material: Neural LiDAR Fields for Novel View Synthesis

Shengyu Huang\textsuperscript{1,2} Zan Gojcic\textsuperscript{2} Zian Wang\textsuperscript{2,3,4} Francis Williams\textsuperscript{2}
Yoni Kasten\textsuperscript{2} Sanja Fidler\textsuperscript{2,3,4} Konrad Schindler\textsuperscript{1} Or Litany\textsuperscript{2}
\textsuperscript{1} ETH Zurich \quad \textsuperscript{2} NVIDIA \quad \textsuperscript{3} University of Toronto \quad \textsuperscript{4} Vector Institute

In this supplementary document, we first present additional information about our dataset and implementation details in Sec. A. We then elaborate on technical details of our methods in Sec. B. Additional results of the two return mask segmentation, more quantitative and qualitative results are provided in Sec. C.

A. Datasets and implementation details

A.1. Dataset

\textbf{Town dataset} To simulate \textit{TownReal} dataset, we approximate a diverged beam profile using 37 subrays and the divergence angle $\gamma_0 = 2 \text{ mrad}$ \cite{2}. We use the subray distribution proposed from \cite{11} (cf. Fig. 1). The dataset is shown in Fig. 2.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Scene & Scene ID \\
\hline
Scene 1 & 10017090168044687777,6380,000,6400,000 \\
Scene 2 & 1009619443888687526,2820,000,2840,000 \\
Scene 3 & 10061305430875486848,1080,000,1100,000 \\
Scene 4 & 10275144660749673822,5755,561,5775,561 \\
\hline
\end{tabular}
\caption{Town dataset.}
\end{table}

\textbf{Waymo dataset} We use the following 4 scenes (cf. Fig. 3) that are mostly static from \textit{Waymo} \cite{7} dataset.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Scene & Scene ID \\
\hline
Scene 1 & 10017090168044687777,6380,000,6400,000 \\
Scene 2 & 1009619443888687526,2820,000,2840,000 \\
Scene 3 & 10061305430875486848,1080,000,1100,000 \\
Scene 4 & 10275144660749673822,5755,561,5775,561 \\
\hline
\end{tabular}
\caption{Waymo dataset.}
\end{table}

A.2. Implementation details

\textbf{Our method.} Our model is implemented based on \textit{torch-ngp} \cite{5, 9} and can be trained on a single RTX 3090 GPU. During training we minimize using the Adam \cite{3} optimiser, with an initial learning rate of 0.005 which linearly decays to 0.0005 towards the end of training. We clip the gradient magnitudes of all parameters to 1.0 to stabilize the optimisation. In the first stage, we sample $N^c = 768$ points and in the second stage $N^f = 64$ points for each ray. The window size $\epsilon$ for volume rendering is set to 0.8 m, and the buffer value $\xi$ between two returns is set to 2 m. The weights in the loss function, i.e., $\lambda_c$, $\lambda_d$, and $\lambda_s$, are set to 50, 0.15, and 0.15, respectively.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Example diverged beam profile approximated via 37 rays.}
\end{figure}

\textbf{LiDARsim.} Because the original implementation is not publicly available, we re-implemented LiDARsim \cite{4} following the paper as close as possible. Specifically, for all points in the training set, we first estimate pointwise normal vectors using all points within a 20 cm radius ball. Then, we apply voxel down-sampling \cite{8} with a voxel size of 4 cm and reconstruct a disk surfel\textsuperscript{1} for each point. Here, the input point represents the disk center and its normal is defined by the estimated normal vector. At inference time, we perform ray-surfel intersection to determine the intersection points. We empirically observed that LiDARsim’s \cite{4} performance is sensitive to the selected surfel radius. Therefore, we have experimented with both a distance-dependent and fixed surfel radius and found that fixed surfel radius of 6 cm and 12 cm for \textit{Waymo} and \textit{Town} dataset, respectively lead to best range accuracy. To enable second range estimation, we augment LiDARsim with a diverged beam profile approximated using 7 rays. To obtain the second return mask, we consider a LiDAR beam to have two returns if the maximum range difference between all subrays is larger than a threshold\textsuperscript{2}. The first return is defined as the closest ray-surfel intersection, while the second return is the nearest one that is at least two meters away. To train the ray drop module, we utilize 40k samples from the Waymo dataset \cite{7}, and only apply this module after the ray-surfel intersection.

\textsuperscript{1}We use the implementation from Point-Cloud-Utils \cite{10} library.

\textsuperscript{2}Sensor-specific parameter, 2 m on \textit{Waymo} dataset.
intersection to refine the ray drop patterns. Please see Fig. 4 for more qualitative results.

Other NeRF methods. We also use torch-ngp [9] codebase to implement other methods, using the same network and sampling configurations as used in ours. To estimate the range, we remove the radiance MLP and instead, apply volume rendering of the sampled $\zeta$ along the ray. For DS-NeRF [1] and URF [6], we replace their positional encoding with a hash-grid [5] to facilitate a fair comparison with i-NGP [5]. Moreover, we substitute the original L2 loss with the L1 loss, as it results in better performance. Finally, we follow the original paper and augment DS-NeRF [1] and URF [6] with the ray distribution loss and line-of-sight loss, respectively, to regularise the underlying geometry.

B. Methodology and loss functions

First range estimation  If the maximum weight at the first stage $w^p_j$ is below a predefined threshold $\eta = 0.1$, we assume that the network is uncertain about the reconstruction and the resulting range estimate may be inaccurate. In these cases, we only apply the coarse stage volume rendering and directly estimate the range as: $\hat{\zeta} = \sum_{j=1}^{N_x} w^c_j \cdot \hat{\zeta}_j$.

Range reconstruction loss For coarse range, we impose a Gaussian distribution around the ground truth $\zeta$ and we anneal the standard deviation $\delta$ during training, the annealing procedure is defined as:

$$\delta_k = \delta_{\text{max}} \left( \delta_{\text{min}} / \delta_{\text{max}} \right)^{k/k_{\text{max}}}$$

where $k$ denotes the iteration number, $k_{\text{max}}$ is the maximum iteration, and $\delta_{\text{max}}$ and $\delta_{\text{min}}$ correspond to empirically determined bounds for the standard deviation. The annealing parameters $\delta_{\text{min}}$ and $\delta_{\text{max}}$ are set to 0.25/0.3 and 1.2/1.6, respectively, for the Town and Waymo datasets. The maximum iteration $k_{\text{max}}$ is set to 16000/24000 for the Town and Waymo datasets. The ground truth weight $\hat{w}_j$ is computed as:

$$\hat{w}_j = \int_{\zeta_j}^{\zeta_{j+1}} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{(x-\hat{\zeta}_j)^2}{2\delta^2} \right) dx.$$  

C. Additional results

Two return mask prediction We conduct an ablation study to investigate different design choices for predicting the two return mask and summarize the results in Tab. 1. We observe that concatenating the range feature $f_{\text{range}}$ with the beam feature $f_{\text{beam}}$ improves the segmentation recall and, consequently, the second range estimation. In addition to predicting the two return mask from the beam feature,
for active sensors, still leads to improved performance, as demonstrated in Tab. 3.

**Qualitative results** We show additional qualitative results in Fig. 6, Fig. 7, Fig. 8, and Fig. 9. We sample the middle frame of each dataset and present the first range errors in range-view projection.

**References**


Figure 2. Visualisation of Town dataset. Employing a diverged beam profile in range simulation results in an overestimation of range in the high range regime (-16 cm to 16 cm). Such range difference is also reflected on delicate structures, as evidenced by the point cloud view.
Figure 3. Visualisations of Waymo dataset. We accumulate all 50 frames for each scene and show their geometry, intensity profile, and sensor positions of training and test sets on Waymo Interp. and Waymo NVS datasets.
Figure 4. Ray drop segmentation on Waymo Interp. dataset using LiDARsim [4]. We show both the initial ray drop mask from ray-surfel query and the refined masks using learned ray-drop model.
Figure 5. Qualitative results of two return mask segmentation.
Figure 6. Qualitative results of first range estimation on TownClean dataset.
Azimuth angle [°]

Figure 7. Qualitative results of first range estimation on TownReal dataset.
Figure 8. Qualitative results of first range estimation on Waymo NVS dataset.
Figure 9. Qualitative results of first range estimation on Waymo Interp. dataset.