Supplementary Material: 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

In this supplementary document, we first present additional information about our dataset, evaluation setting, 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

Town dataset To simulate *TownReal* dataset, we approximate a diverged beam profile using 37 subrays and the divergence angle $\gamma_0 = 2$ mrad [3]. We use the subray distribution proposed from [14] (*cf*. Fig. 1). The dataset is shown in Fig. 3.

Waymo dataset We use the following 4 scenes (*cf*. Fig. 4) that are mostly static from *Waymo* [10] dataset

	Scene ID
Scene 1	10017090168044687777_6380_000_6400_000
Scene 2	10096619443888687526_2820_000_2840_000
Scene 3	10061305430875486848_1080_000_1100_000
Scene 4	10275144660749673822_5755_561_5775_561

A.2. Evaluation setting

***Waymo NVS* setting** We simulate the new trajectory by shifting the sensor by [1.5, 1.5, 0.5] meters (see Fig. 4), yielding an overall displacement of ≈2.18 meters. This displacement magnitude corresponds to the requirements of various tasks, such as lane changes or adapting the sensor rig from a car to a truck. Moreover, our displacement from the trajectory is similar [15] or even larger [8] than used in prior NVS works. Nevertheless, we run additional experiments by varying the displacements and report results in Tab. 1. NFL consistently outperforms baseline methods under different settings, and the improvement is more pronounced under large displacements.

	i-NGP	DS-NeRF	URF	LiDARsim	Ours
(0.5, 0.5, 0.5)	7.0 / 14.4	7.0 / 16.0	9.0 / 19.6	16.1 / 33.1	**5.4 / 13.0**
(1.5, 1.5, 1.0)	8.4 / 17.6	7.8 / 18.5	11.0 / 27.5	16.5 / 37.9	**5.8 / 14.3**
(2.5, 2.5, 1.5)	11.6 / 28.0	9.3 / 22.8	13.9 / 35.5	17.2 / 46.3	**6.4 / 18.4**

Table 1. Varying the displacement on *Waymo NVS* dataset. Numbers are reported as *MedAE* / *CD* [cm].

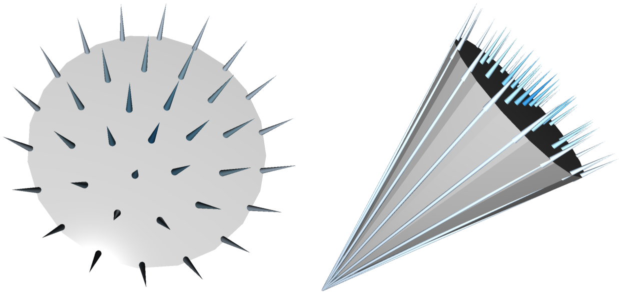


Figure 1. Example diverged beam profile approximated via 37 diverged rays.

Point cloud registration task We utilize 49 paired consecutive frames per scene, with a relative displacement of ≈1 meter. *TE* is reported in centimeters and *RE* is reported in degrees.

A.3. Implementation details

Our method. Our model is implemented based on *torch-ngp* [7, 12] and can be trained on a single RTX 3090 GPU. During training we minimize using the Adam [4] 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 ϵ for volume rendering is set to 0.8 m, and the buffer value ξ between two returns is set to 2 m. The weights in the loss function, i.e., λ_e, λ_d, and λ_s, are set to 50, 0.15, and 0.15, respectively.

LiDARsim. Because the original implementation is not publicly available, we re-implemented LiDARsim [6] 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 [11] with a voxel size of 4 cm and reconstruct a disk surfel[1] for each point. Here, the input point represents the disk center and it orientation 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 [6] 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 *Waymo* and *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[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 [10], and only apply this module after the ray-surfel intersection to refine the ray drop patterns. Please see Fig. 5 for more qualitative results.

Other NeRF methods. We also use *torch-ngp* [12] 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 ζ along the ray. For DS-NeRF [2] and URF [9], we replace their positional encoding with a hash-grid [7] to facilitate a fair comparison with i-NGP [7]. 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 [2] and URF [9] 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^c 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: $\zeta = \sum_{j=1}^{N^c} w_j^c \cdot \zeta_j$.

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

$$\delta_k = \delta_{\max} \left(\frac{\delta_{\min}}{\delta_{\max}} \right)^{k/k_{\max}} \tag{1}$$

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

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

C. Additional results

Runtime analysis Our *central ray* version takes 4.1 ms per frame to render the single returns on an RTX 3090 GPU, while other NeRF-style methods require 2.4 ms. Only around 10% of rays have second returns, resulting in low computational overhead. While our *diverged beam* incurs additional costs due to querying diverged rays, it can be disabled if needed, without compromising first return performance (*cf*. Tab. 1). Our re-implementation of LiDARsim achieves 10 Hz runtime, but could be further improved using accelerated ray-tracing, *e.g*. OptiX. Note that all methods already match or even (greatly) exceed the normal LiDAR measurement frequency (≈10 Hz).

Ray drop modelling There clearly is a link between ray drops and beam divergence. However, we found that modeling it through the beam feature yields worse performance, possibly because $\mathbf{f}_{\text{beam}}$ uses $\mathbf{f}_{\text{range}}$, which encodes the statistics of returns and is less meaningful for dropped rays. In future work, beam divergence could instead be incorporated through Intergrated Positional Encoding [1] to model ray drops.

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. 2. We observe that concatenating the range feature $\mathbf{f}_{\text{range}}$ with the beam feature $\mathbf{f}_{\text{beam}}$ improves the segmentation recall and, consequently, the second range estimation. In addition to predicting the two return mask from the beam feature, we experiment with a simple heuristic-based baseline that thresholds the depth standard deviation of sub-rays. Specifically, we considered a LiDAR beam to have two returns if

[1] We use the implementation from Point-Cloud-Utils [13] library.

[2] Sensor-specific parameter, 2 m on *Waymo* dataset.

Features			Two return segmentation			Second range		
$\bar{\mathbf{f}}_{\text{geo}}$	$\mathbf{f}_{\text{dir}}$	$\mathbf{f}_{\text{range}}$	Recall ↑	Precision ↑	IoU ↑	Recall@0.5 ↑	MAE ↓	MedAE ↓
✓			78.0	61.6	52.8	60.1	620.1	26.7
✓	✓		79.8	62.9	54.5	61.1	589.1	21.8
✓	✓	✓	82.1	55.6	49.8	67.4	505.1	13.4
threshold depth std.			30.8	24.2	14.8	24.7	1532.2	1461.4

Table 2. Qualitative results of two return segmentation on *Waymo Interp.* dataset.

	Vehicle			Background		
Method	Recall ↑	Precision ↑	IoU ↑	Recall ↑	Precision ↑	IoU ↑
i-NGP [7] + L2	71.1	**97.0**	69.4	**99.6**	96.5	96.2
i-NGP [7]	94.8	89.7	85.6	98.7	99.4	98.1
DS-NeRF [2]	91.4	88.9	82.2	98.7	99.1	97.8
URF [9]	93.8	89.0	84.1	98.6	99.3	97.9
Lidarsim [6]	92.2	74.4	70.2	95.9	99.1	95.1
Ours	**95.7**	91.2	**87.6**	98.8	**99.5**	**98.3**

Table 3. Semantic segmentation results 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 [7] + L2	63.6	14.8	37.1	78.2	18.4	44.5	41.4	14.7	24.9	47.3	17.6	29.5
i-NGP [7]	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 [2]	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 [9]	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 [6]	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 4. Results of LiDAR novel view synthesis for the first range.

	TownClean			Waymo Interp.		
Method	MAE ↓	MedAE ↓	CD ↓	MAE ↓	MedAE ↓	CD ↓
i-NGP [7] + L2	60.8 (-2.8)	12.6 (-2.2)	34.4 (-2.7)	40.8 (-0.6)	13.1 (-1.6)	24.0 (-0.8)
i-NGP [7]	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 [2]	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 [9]	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 5. Ablation study of volume rendering for active sensing.

	TownClean			TownReal			Waymo NVS		
Method	Rec@5 ↑	RE ↓	TE ↓	Rec@5 ↑	RE ↓	TE ↓	Rec@2 ↑	RE ↓	TE ↓
i-NGP [7] + L2	40.6	0.2	6.2	39.6	0.2	6.7	26.5	0.1	3.2
i-NGP [7]	70.3	0.1	4.2	76.0	0.1	4.2	60.2	0.1	1.9
DS-NeRF [2]	58.3	0.2	5.1	56.2	0.2	5.1	42.3	0.1	2.4
URF [9]	61.5	0.2	5.0	59.9	0.1	4.7	32.1	0.1	2.7
LiDARsim [6]	**82.8**	**0.1**	**3.4**	79.2	**0.1**	**3.4**	62.8	**0.1**	1.8
Ours	80.2	0.1	3.7	**85.9**	0.1	**3.4**	**71.9**	**0.1**	**1.7**

Table 6. Point cloud registration results on three datasets.

	Vehicle			Background		
Method	Recall ↑	Precision ↑	IoU ↑	Recall ↑	Precision ↑	IoU ↑
i-NGP [7] + L2	68.4	**90.2**	64.1	**99.3**	96.3	95.6
i-NGP [7]	93.2	85.9	80.9	98.3	99.2	97.6
DS-NeRF [2]	90.7	87.1	80.2	**98.5**	98.9	97.4
URF [9]	87.8	81.7	73.7	98.0	98.4	96.5
Lidarsim [6]	90.5	70.5	65.9	94.9	99.0	94.0
Ours	**95.9**	87.0	**83.9**	98.3	**99.5**	**97.8**

Table 7. Semantic segmentation results on *Waymo NVS* dataset.

the standard deviation is above 30[3] cm. However, as shown in Table 2, this approach achieves limited success and performs much worse than the learned methods. More qualitative results are presented in Fig. 6.

Importance of the second return Multiple returns are critical for vegetation analysis in remote sensing [5]. NFL is the first work to model the second return by combining beam divergence and *truncated* volume rendering. Unfortunately, second returns do not have semantic annotations in the Waymo dataset, which precluded a quantitative analysis. Nevertheless, qualitatively the rendered second returns are located mostly in vegetation regions, as shown in Fig. 2. This correlation suggests that secondary returns could indeed be useful for detecting vegetation.

Semantic segmentation on *Waymo Interp.* dataset We report additional semantic segmentation results on *Wamo Interp.* dataset in Tab. 3. NFL achieves the best performance for vehicle segmentation. Please note that *Waymo Interp.* is of significantly smaller size (10 test frames *vs*. 50 frames per scene in other datasets).

Quantitative results We perform further experiments to evaluate an additional baseline method, denoted as *i-NGP [7] + L2*, which optimizes the range estimation through L2 loss [2, 9]. The comprehensive results of our experimentation are presented in Tab. 4, Tab. 5, Tab. 6, and Tab. 7. Our findings reveal that the L2 loss performs inferior to its L1 loss counterpart (*i.e.* i-NGP [7]). However, replacing the standard volume rendering with the proposed formulation for active sensors, still leads to improved performance, as demonstrated in Tab. 5.

Qualitative results We show additional qualitative results in Fig. 7, Fig. 8, Fig. 9, and Fig. 10. We sample the middle frame of each dataset and present the first range errors in range-view projection.

[3] Empirically determined as it leads to the best Intersection-of-Union score.

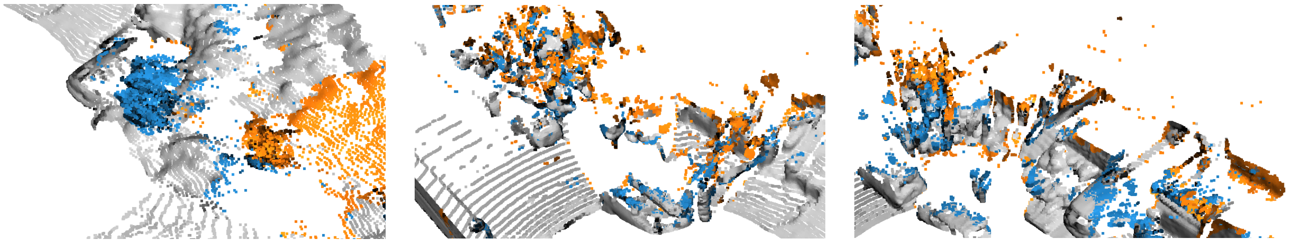


Figure 2. Rendered secondary returns are color-coded in yellow.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV*, 2023.

[2] 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.

[3] Craig Glennie. Calibration and kinematic analysis of the Velodyne HDL-64E S2 lidar sensor. *Photogrammetric Engineering & Remote Sensing*, 78(4):339–347, 2012.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[5] Kevin Lim, Paul Treitz, Michael Wulder, Benoît St-Onge, and Martin Flood. Lidar remote sensing of forest structure. *Progress in physical geography*, 27(1):88–106, 2003.

[6] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. LiDARsim: Realistic LiDAR simulation by leveraging the real world. In *CVPR*, 2020.

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

[8] Julian Ost, Issam Laradji, Alejandro Newell, Yuval Bahat, and Felix Heide. Neural point light fields. In *CVPR*, 2022.

[9] 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.

[10] 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.

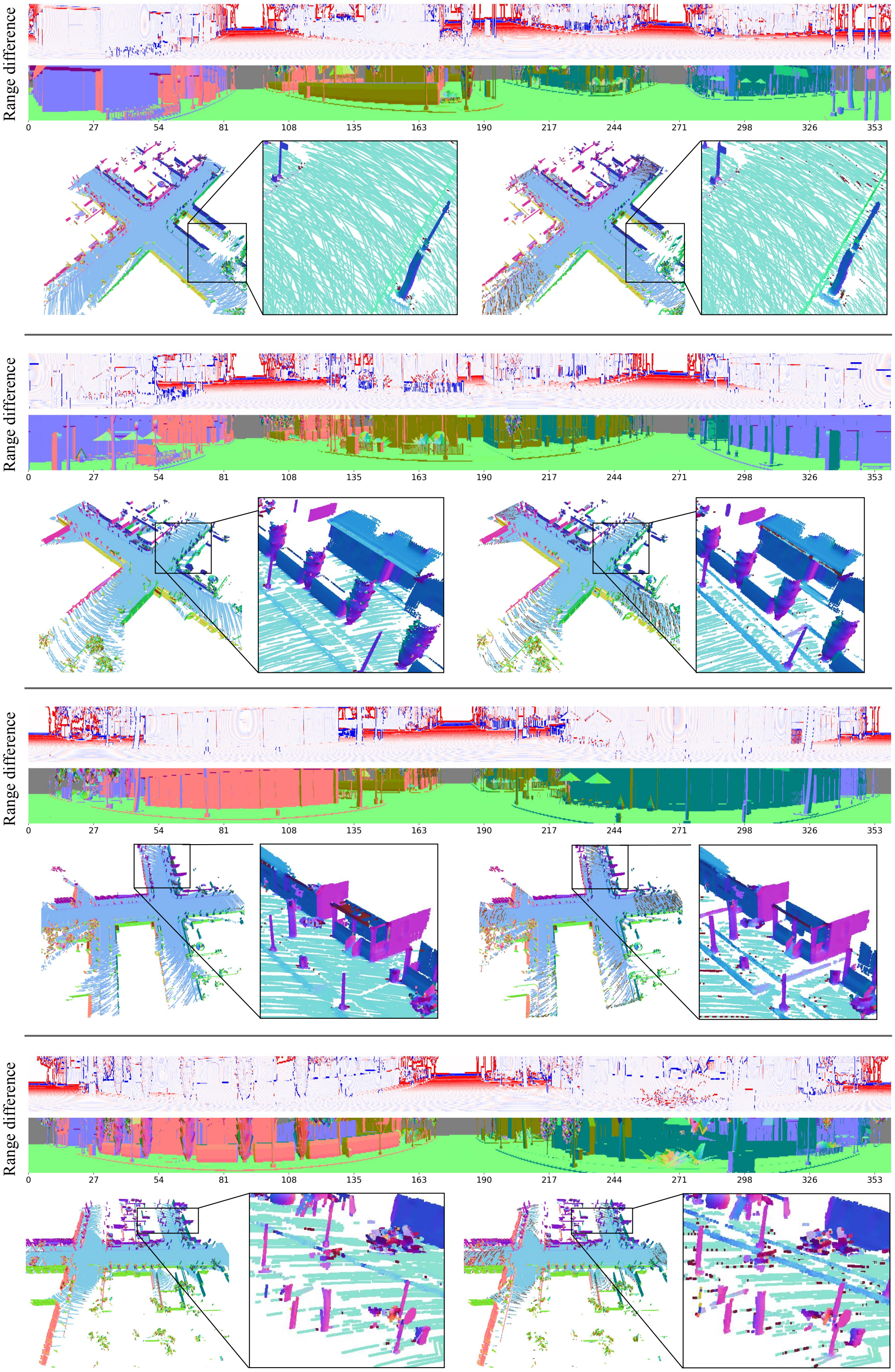
[11] Haotian Tang, Zhijian Liu, Xiuyu Li, Yujun Lin, and Song Han. TorchSparse: Efficient Point Cloud Inference Engine. In *Conference on Machine Learning and Systems (MLSys)*, 2022.

[12] Jiaxiang Tang. Torch-ngp: a PyTorch implementation of instant-ngp, 2022. https://github.com/ashawkey/torch-ngp.

[13] Francis Williams. Point cloud utils, 2022. https://www.github.com/fwilliams/point-cloud-utils.

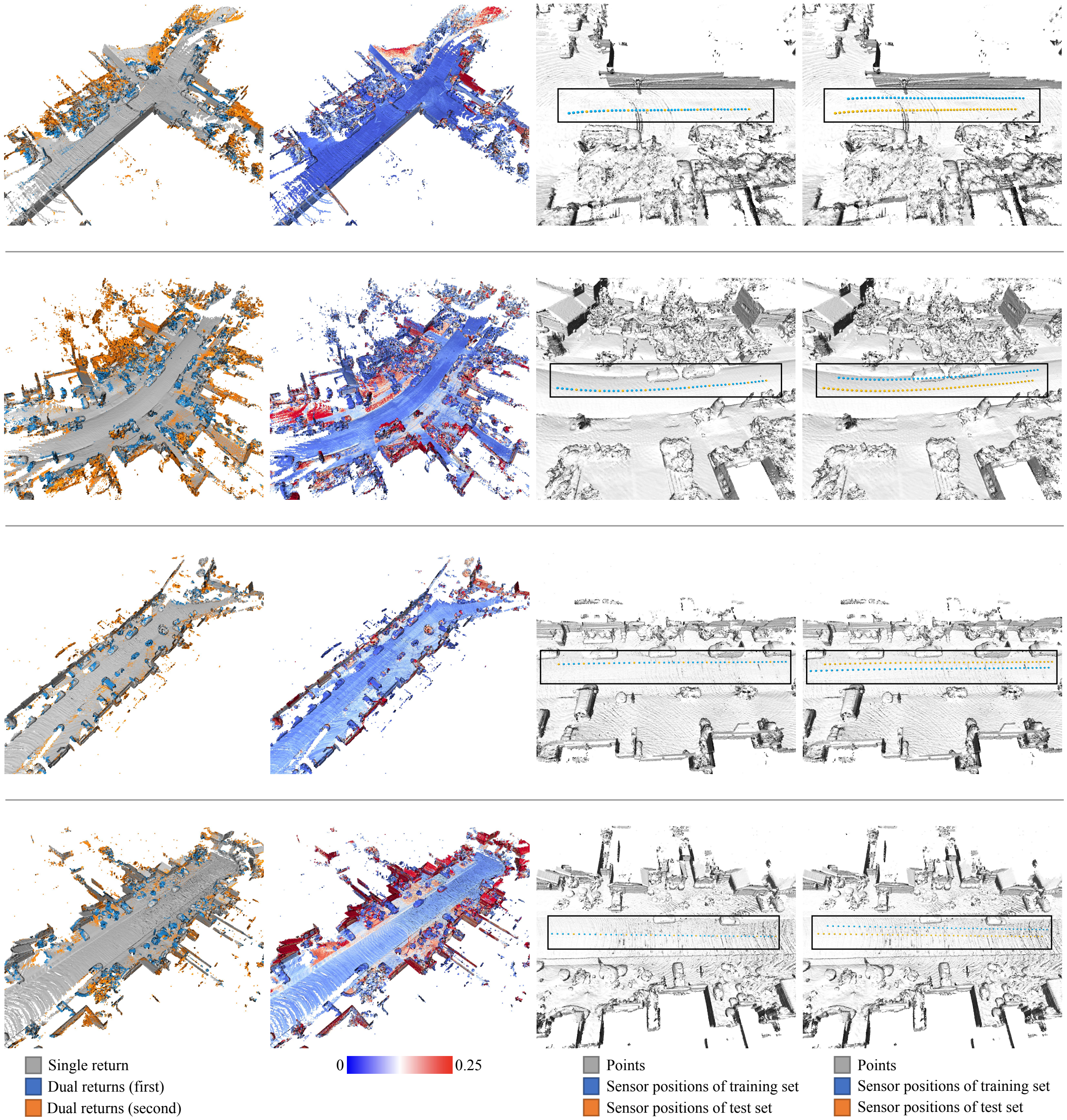
[14] Lukas Winiwarter, Alberto Manuel Esmorís Pena, Hannah Weiser, Katharina Anders, Jorge Martínez Sánchez, Mark Searle, and Bernhard Höfle. Virtual laser scanning with HELIOS++: A novel take on ray tracing-based simulation of topographic full-waveform 3d laser scanning. *Remote Sensing of Environment*,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

[15] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *CVPR*, 2023.



TownClean *TownReal*

Figure 3. Visualisation of *Town* dataset. Employing a diverged beam profile in range simulation results in an overestimation of range in the high range regime (-16 16 cm). Such range difference is also reflected on delicate structures, as evidenced by the point cloud view.


⬜ Single return
🟦 Dual returns (first)
🟧 Dual returns (second)
0
0.25
⬜ Points
🟦 Sensor positions of training set
🟧 Sensor positions of test set
⬜ Points
🟦 Sensor positions of training set
🟧 Sensor positions of test set

(a) Geometry (b) Intensity (c) *Waymo Interp.* (d) *Waymo NVS*

Figure 4. 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.

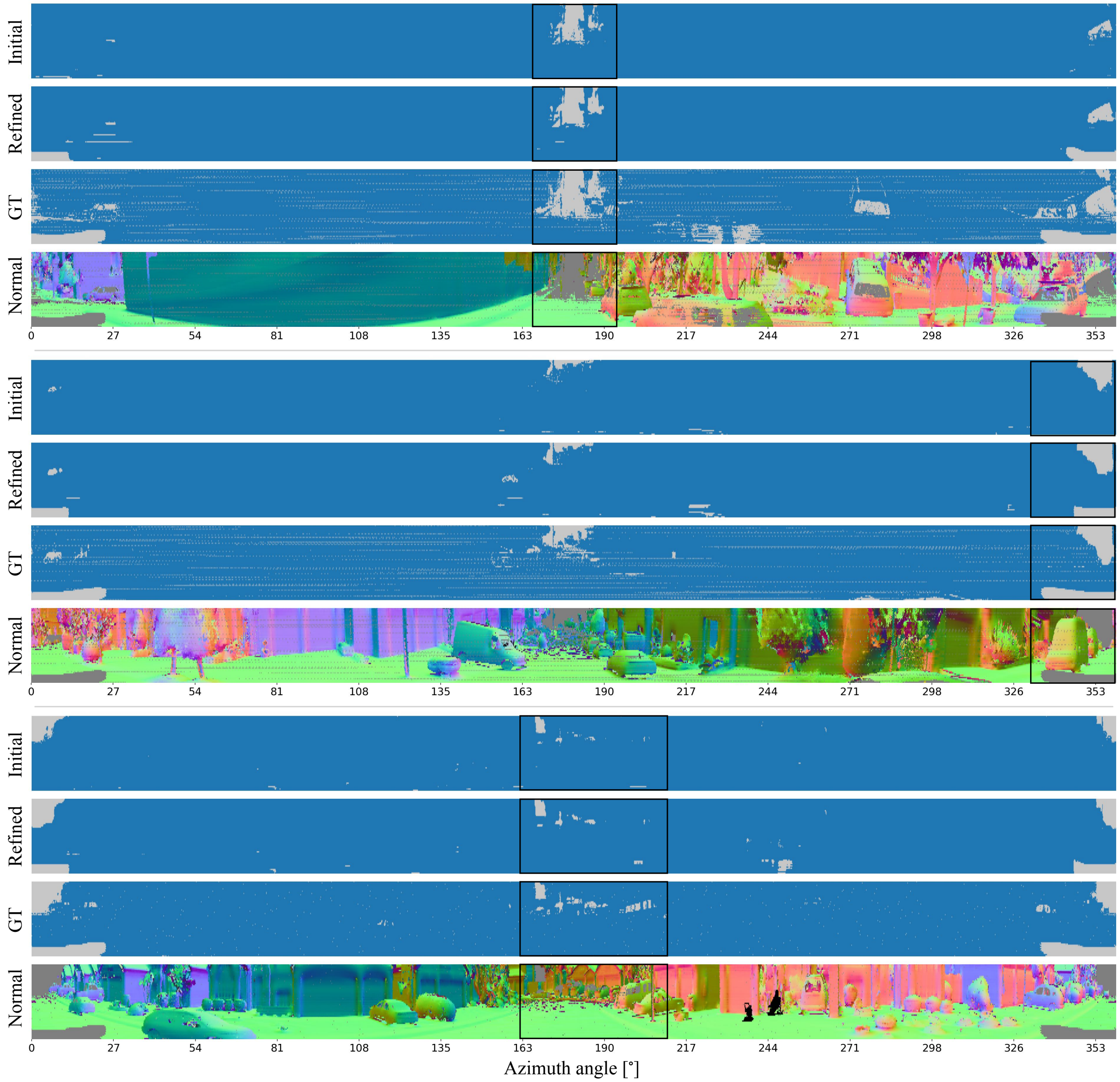


Figure 5. Ray drop segmentation on *Waymo Interp.* dataset using LiDARsim [6]. We show both the initial ray drop mask from ray-surfel query and the refined masks using learned ray-drop model.

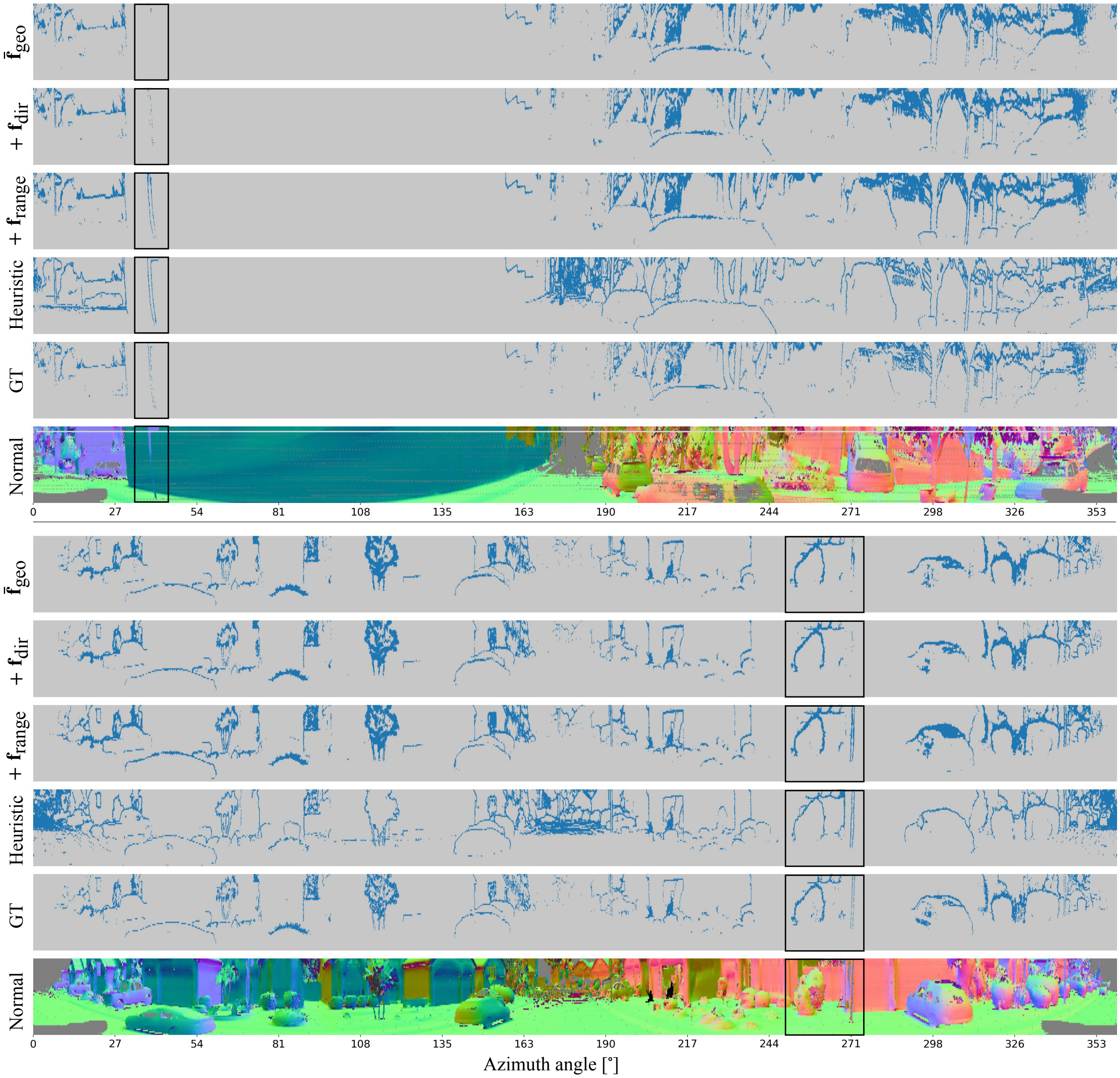


Figure 6. Qualitative results of two return mask segmentation.

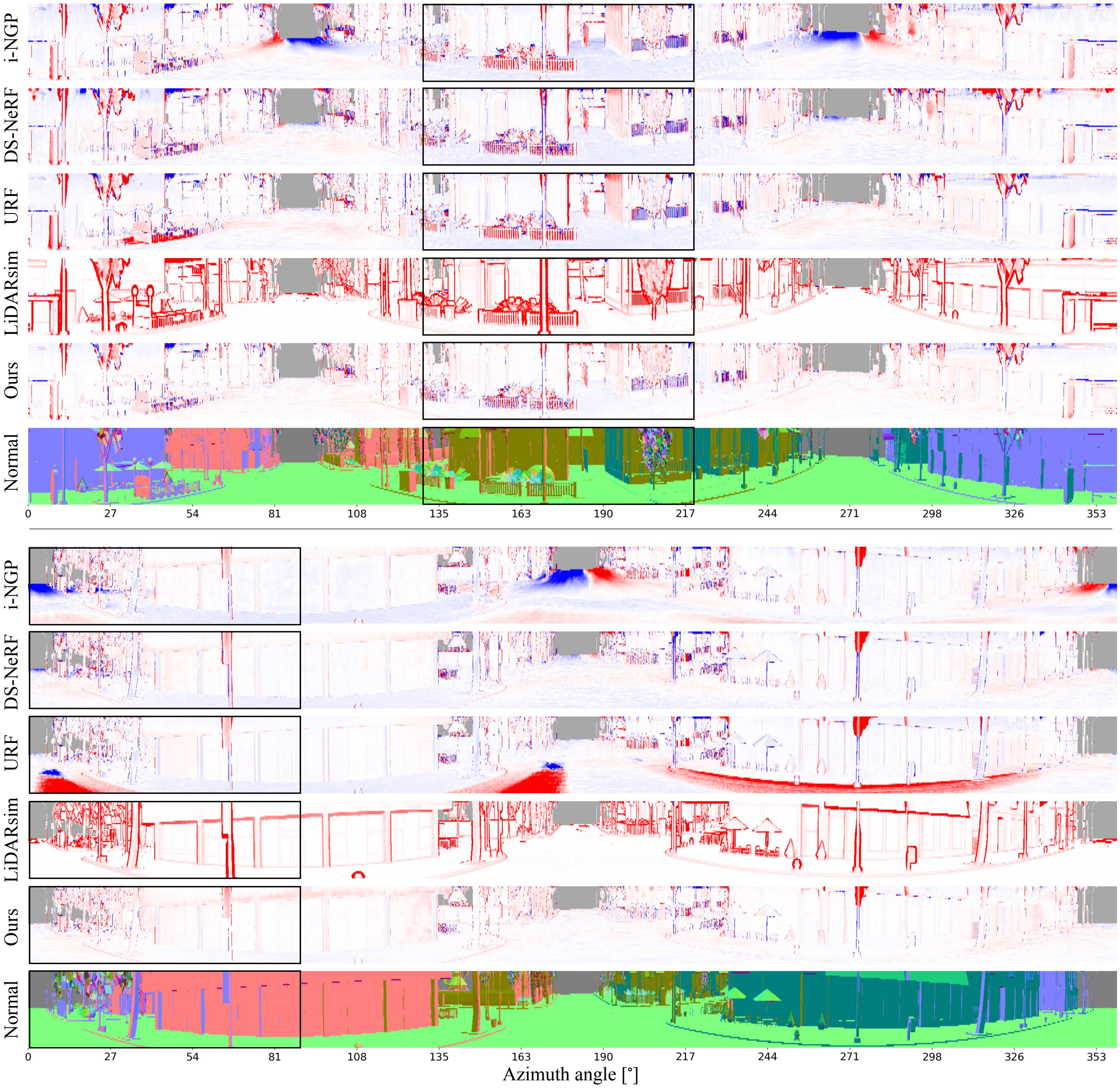


Figure 7. Qualitative results of first range estimation on *TownClean* dataset.

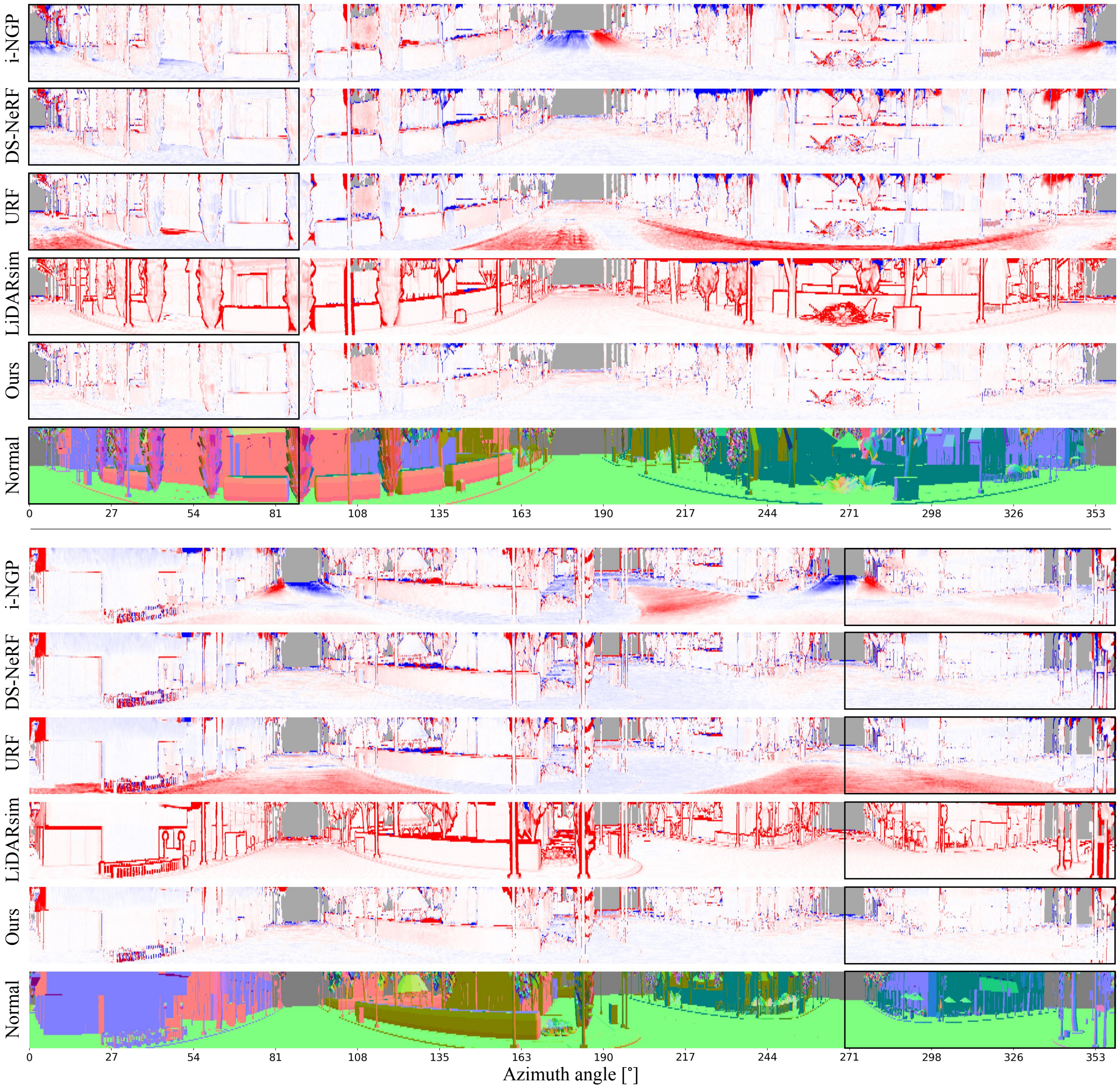


Figure 8. Qualitative results of first range estimation on *TownReal* dataset.

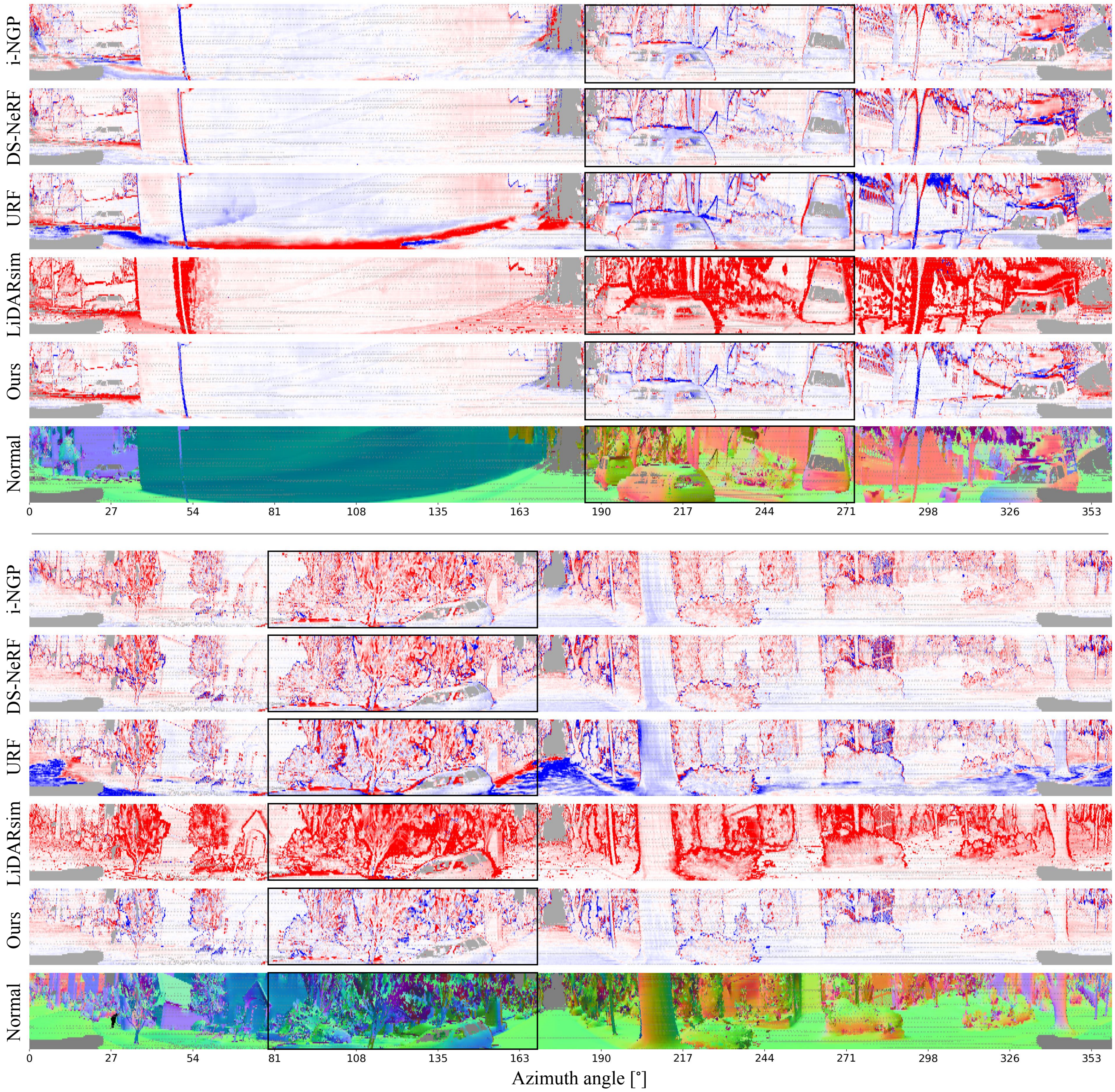


Figure 9. Qualitative results of first range estimation on *Waymo NVS* dataset.

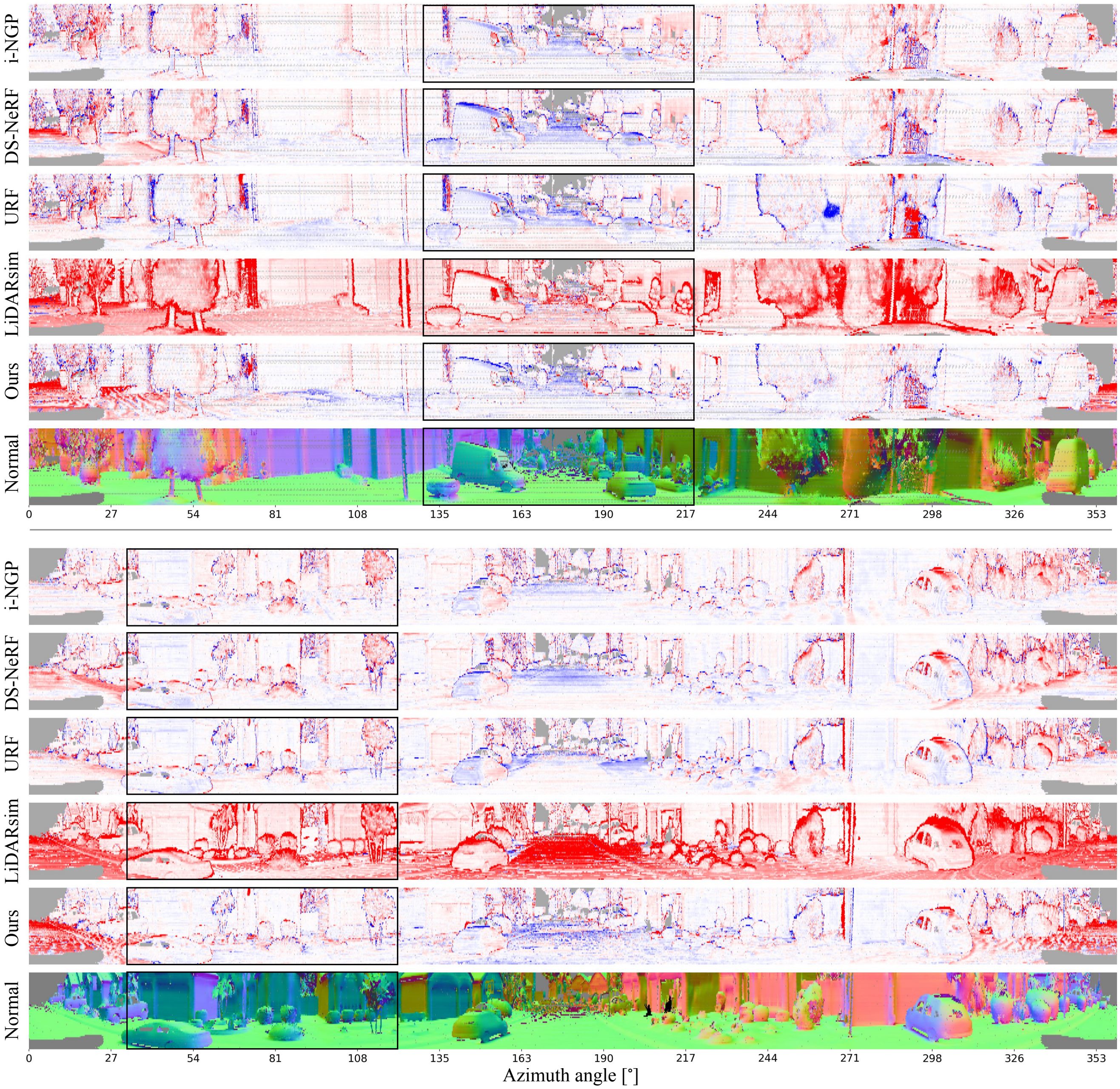


Figure 10. Qualitative results of first range estimation on *Waymo Interp.* dataset.