

# NeuRadar: Neural Radiance Fields for Automotive Radar Point Clouds

## Supplementary Material

In the supplementary material, we present details of NeuRadar’s implementation, more information about ZOD’s radar data, and additional visualizations. In Appendix A, we provide information about the training process, network architecture, and losses. In addition, we provide our hyperparameter values. In Appendix B, we present more detailed information about the sequences used to evaluate our network and ZOD’s radar point cloud data. In Appendix C.1, we investigate the effects of adding radar point cloud data on image and lidar reconstruction by comparing our method to NeuRAD, followed by an exploration of using a DETR-like object detection network as a radar decoder. We then provide more figures depicting our results in Appendix C.3. Finally, we identify the limitations of our method and outline potential directions for future work in Appendix D.

### A. Implementation Details

**Training:** We train all parts of our model jointly with 20,000 iterations, using the Adam optimizer. Regarding the number of rays in each iteration, we follow the settings in NeuRAD [44] for camera and lidar, *i.e.*, 16,384 lidar rays and 40,960 camera rays. For radar, the number of rays in each iteration is not fixed. Instead, the number of radar rays in each iteration equals the number of radar rays in each scan multiplied by the number of radar scans loaded in each iteration. The radar specifications for the ZOD and VoD are shown in Tab. 4. The number of rays per iteration is 54,784 for ZOD and 70,400 for and VoD.

For the optimization, we adopt the same settings for existing modules in NeuRAD. For the new module, the radar decoder, we use a warmup of 5,000 steps and a learning rate of 0.001 that decays by an order of magnitude throughout the training.

**Neural Feature Field:** We use NeuRAD’s hyperparameter settings for the neural feature field (NFF) in NeuRadar. Additionally, radar rays have specific hyperparameters, such as ray divergence and a scaling parameter, which are explained in Sec. 4.1.1. All NFF-related settings are listed in Tab. 5.

**Hashgrids:** We follow NeuRAD and employ the efficient hashgrid implementation provided by tiny-cuda-nn [33], configuring two distinct hashgrids, one for the static environments and one for dynamic actors. For the static environments, a significantly larger hash table is employed, given that actors comprise only a minimal area of the overall scene.

**Losses:** In Sec. 3.3 and Sec. 4.2, we explain that the total loss in NeuRadar comprises the radar loss  $\mathcal{L}^{\text{radar}}$ , image loss  $\mathcal{L}^{\text{image}}$ , and lidar loss  $\mathcal{L}^{\text{lidar}}$ , with the latter two form-

ing the NeuRAD loss. While the paper focuses on the theory and motivation behind these losses, we present their detailed equations here.

The image loss is computed by

$$\mathcal{L}^{\text{image}} = \frac{1}{N_p} \sum_{i=1}^{N_p} (\lambda^{\text{rgb}} \mathcal{L}_i^{\text{rgb}} + \lambda^{\text{vgg}} \mathcal{L}_i^{\text{vgg}}), \quad (15)$$

where  $N_p$  denotes the number of patches, and  $\lambda^{\text{rgb}}$  and  $\lambda^{\text{vgg}}$  are weighting hyperparameters. The lidar loss is defined as

$$\mathcal{L}^{\text{lidar}} = \frac{1}{N} \sum_{i=1}^N (\lambda^d \mathcal{L}_i^d + \lambda^{\text{int}} \mathcal{L}_i^{\text{int}} + \lambda^{p_d} \mathcal{L}_i^{p_d} + \lambda^w \mathcal{L}_i^w), \quad (16)$$

where  $N$  denotes the number of lidar points, and  $\lambda^d$ ,  $\lambda^{\text{int}}$ ,  $\lambda^{p_d}$ , and  $\lambda^w$  are weighting hyperparameters. Finally, the radar loss is calculated by

$$\mathcal{L}^{\text{radar}} = \begin{cases} \lambda^{\text{radar}} \mathcal{L}_{\text{det}}^{\text{radar}}, & \text{if deterministic modeling,} \\ \lambda^{\text{radar}} \mathcal{L}_{\text{prob}}^{\text{radar}}, & \text{if probabilistic modeling,} \end{cases} \quad (17)$$

where  $\lambda^{\text{radar}}$  is a weighting hyperparameter. The values of these hyperparameters are given in Tab. 5.

**Offset Head:** A critical hyperparameter for offset prediction is the maximum offset value, which acts as a constraint on the radar decoder’s offset head. With respect to this hyperparameter, a small value makes the estimations from NFF dominant, while a large value gives the radar decoder more flexibility. Tab. 6 presents the performance for varying maximum offset values. The experiments are conducted on our probabilistic radar model across three evaluation sequences from two datasets. For ZOD, a maximum offset of 1.5 meters yields good results in terms of both CD and EMD. However, the best CD and EMD values for VoD do not align. Averaging the metrics suggests that 1.5 meters is also suitable for VoD.

### B. Datasets

In this section, we provide more detailed information about the datasets used to evaluate our network.

#### B.1. ZOD

We used ZOD sequences 000030, 000546, and 000811 for our ablation studies and hyperparameter tuning. For our final experiments we used the following ten sequences: 000005, 000221, 000231, 000244, 000387, 000581, 000619, 000657, 000784, 001186. These sequences vary in ego speed, lighting conditions (both day

Table 4. Specifications for the radar in two datasets. The unit for ray divergence is radians.

Dataset	Azimuth range	Elevation range	Ray divergence	#rays per scan	#scans	#rays per iteration
ZOD	$\pm 45.84^\circ$	$(-4.58^\circ, 22.92^\circ)$	0.015(0.8594 $^\circ$ )	3424	16	54784
VoD	$\pm 57.29^\circ$	$(-22.34^\circ, 28.07^\circ)$	0.02(1.14 $^\circ$ )	4400	16	70400

Table 5. Hyperparameters for NeuRadar. The hyperparameter values are universal across the three datasets, except for the radar ray divergence. The most suitable value for this hyperparameter is 0.0125 for ZOD and VoD and 0.025 for nuScenes.

	Hyperparameter	Value
Neural feature field	RGB upsampling factor	3
	proposal samples	128, 64
	SDF $\beta$	20.0 (learnable)
	power function $\lambda$	-1.0
	power function scale	0.1
	appearance embedding dim	16
	hidden dim (all networks)	32
	NFF feature dim	32
	Radar ray divergence $\delta_\varphi$ and $\delta_\theta$	0.0125/0.025
	Radar ray scaling parameter $\zeta$	$\frac{1}{16}$
Hashgrids	hashgrid features per level	4
	actor hashgrid levels	4
	actor hashgrid size	$2^{15}$
	static hashgrid levels	8
	static hashgrid size	$2^{22}$
	proposal features per level	1
	proposal static hashgrid size	$2^{20}$
proposal actor hashgrid size	$2^{15}$	
Loss weights	$\lambda^{\text{rgb}}$	5.0
	$\lambda^{\text{vgg}}$	5e-2
	$\lambda^{\text{int}}$	1e-1
	$\lambda^{\text{d}}$	1e-2
	$\lambda^{\text{w}}$	1e-2
	$\lambda^{P_d}$	1e-2
	proposal $\lambda^{\text{d}}$	1e-3
	proposal $\lambda^{\text{w}}$	1e-3
	interlevel loss multiplier	1e-3
	$\lambda^{\text{radar}}$	2e-2
Learning rates	actor trajectory lr	1e-3
	cnn lr	1e-3
	camera optimization lr	1e-4
	transformer lr	1e-3
	remaining parameters lr	1e-2

and night), weather conditions (sunny, snowy, and cloudy), and scenario type (highway, city, residential), which we deemed appropriate for our experiments.

The radar point clouds in ZOD are captured every 60 ms and stored in a standard binary file format (.npy) for each

Table 6. Results for various maximum offsets in the Cartesian coordinate system. The unit of offset is meters.

Max Offset	ZOD		VoD	
	CD ↓	EMD ↓	CD ↓	EMD ↓
1.0	4.68	6.05	4.19	4.48
1.5	<b>3.92</b>	<b>4.94</b>	4.02	<b>4.28</b>
2.0	4.54	6.22	4.07	4.28
2.5	4.62	6.21	4.07	4.42
3.0	4.55	6.14	<b>3.99</b>	4.33

ZOD *Sequence* and *Drive*. The data contains timestamps in UTC, radar range in meters, azimuth and elevation angles in radians, range rate in meters per second, amplitude (or SNR), validity, mode, and quality. The radar switches between three modes depending on the ego vehicle speed, and the sensor has a different maximum detection range in each mode. Mode 0 represents the radar point clouds captured when the vehicle speed is less than 60 to 65 kph with a maximum detection range of 102 meters, while modes 1 and 2 represent vehicle speeds of between 60 to 65 kph and 110 to 115 kph, and more than 110 to 115 kph, respectively, with maximum detection ranges of 178.5 and 250 meters. The azimuth angle values are between -50 and 50 degrees. The quality value also changes from 0 to 2, with 2 indicating the highest quality for the detections. The radar extrinsic calibration information (i.e., latitude, longitude, and angle) is provided in calibration files, indicating its position relative to the reference coordinate frame.

## B.2. VoD

VoD contains driving scenarios captured at 10 Hz around Delft city from the university campus, the suburbs, and the old town with many pedestrians and cyclists, so the dataset itself is not very diverse and is rather challenging for NeRF-based methods. Since VoD does not provide sequence numbers, we created a set of "sequences", each of which is roughly 30 seconds and originates from a different drive in the dataset. We provide the range of frames numbers in the dataset for each sequence. The VoD sequences used for ablations and hyperparameter tuning were 1850-2150, 7600-7900, and 8482-8749. For our final experiments, we used 100-400, 2220-2520, 2532-2798, 2900-3200, 3277-3575, 3650-3950, 4050-4350, 4387-4652, 4660-4960, and 6800-7100.

## C. Additional Results

### C.1. Effects on Image and Lidar Rendering

To investigate whether incorporating a radar branch affects image and lidar rendering, we compare the performance of NeuRAD and NeuRadar on image and lidar rendering tasks. The results are shown in Tab. 7. We report PSNR, SSIM [45], and LPIPS [56] as image similarity metrics for camera simulation. We evaluate the fidelity of lidar simulation using four metrics: L2 median depth error, RMSE intensity error, ray drop accuracy, and Chamfer Distance (CD). For both datasets NeuRadar’s performance is similar to NeuRAD and the addition of radar does not affect the camera and lidar reconstruction performance.

### C.2. DETR as Radar Decoder

As mentioned in Sec. 4.1.2, a potential solution to radar modeling is to directly predict radar detections, or in this case MB parameters, from the NFF features using a network commonly used for object detection. To further explore this idea, we evaluate novel view synthesis using a DETR-like transformer network [9] as the radar decoder, where the number of output queries is equal to the maximum potential number of radar detections. In Fig. 6 we show the radar and lidar detections generated by this network. Although the method can predict a reasonable point cloud from previously seen viewpoints, it is completely incapable of true novel view synthesis, implying that the network ignores the geometric info about the scene and has merely learned to copy the ground truth regardless of features.

### C.3. Visualizations

In this section, we visualize the qualitative results of our experiments. Fig. 7 shows the novel view synthesis results for two VOD sequences using our probabilistic method. VOD is a dataset specifically curated for urban scenarios with many pedestrians and cyclists, making it a challenging dataset to use for NeRF-based methods.

## D. Limitations and Future Work

In this work, NeuRadar effectively generates realistic radar point clouds. However, certain characteristics of radar data are not fully captured. Here, we describe two limitations of our work, which also point to directions for enhancing NeuRadar.

First, NeuRadar’s radar decoder does not predict radar range rate, signal-to-noise ratio (SNR), or radar cross section (RCS). We believe that extending NeuRadar to incorporate these values and addressing the associated challenges could make a valuable contribution. Another limitation stems from the inherent mechanisms of NeRFs. Designed primarily for visual data, NeRFs focus on surface geometry and visible features, which hinders NeuRadar’s

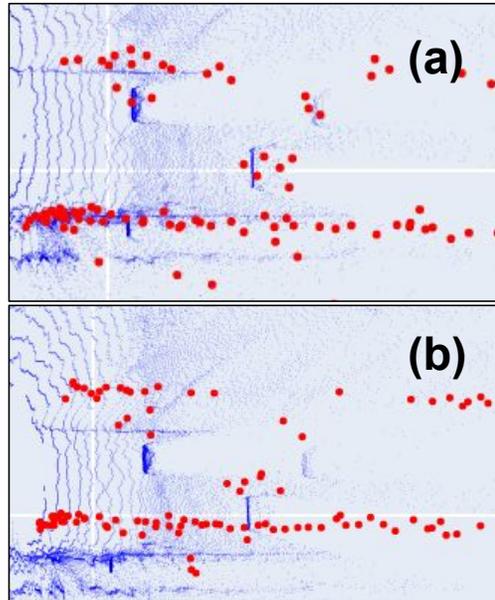


Figure 6. Radar point cloud (in red) rendered by the naive DETR-based radar decoder depicted along rendered lidar point cloud. (a) shows the output for an interpolated sensor pose in a ZOD sequence, and (b) shows the radar detections rendered with a 2 meter ego pose shift. The radar detections have merely shifted in position and do not reflect the geometry of the scene as shown by lidar.

ability to fully leverage radar’s strength in detecting visually occluded objects. Addressing this limitation presents a promising research direction.

Table 7. Performance comparison of novel view synthesis for image and lidar. NeuRAD results are obtained using its public code with recommended settings.

Dataset	Method	Camera			Lidar			
		PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Depth $\downarrow$	Intensity $\downarrow$	Drop acc. $\uparrow$	Lidar CD $\downarrow$
ZOD	NeuRAD	<b>30.9</b>	<b>0.878</b>	<b>0.187</b>	<b>0.028</b>	<b>0.041</b>	<b>95.8</b>	3.69
	Baseline	29.96	0.871	0.192	0.035	0.043	95.5	3.75
	Deterministic	30.4	0.870	0.190	0.035	0.045	95.6	3.73
	Probabilistic	30.1	0.870	0.191	0.030	<b>0.041</b>	95.5	<b>3.59</b>
VoD	NeuRAD	21.68	<b>0.687</b>	0.366	<b>0.167</b>	<b>0.158</b>	85.78	13.96
	Baseline	21.59	0.680	0.372	0.217	0.158	85.73	14.45
	Deterministic	21.63	0.683	0.365	0.172	0.163	<b>85.86</b>	11.92
	Probabilistic	<b>21.75</b>	0.683	<b>0.363</b>	<b>0.167</b>	0.159	85.75	<b>11.66</b>

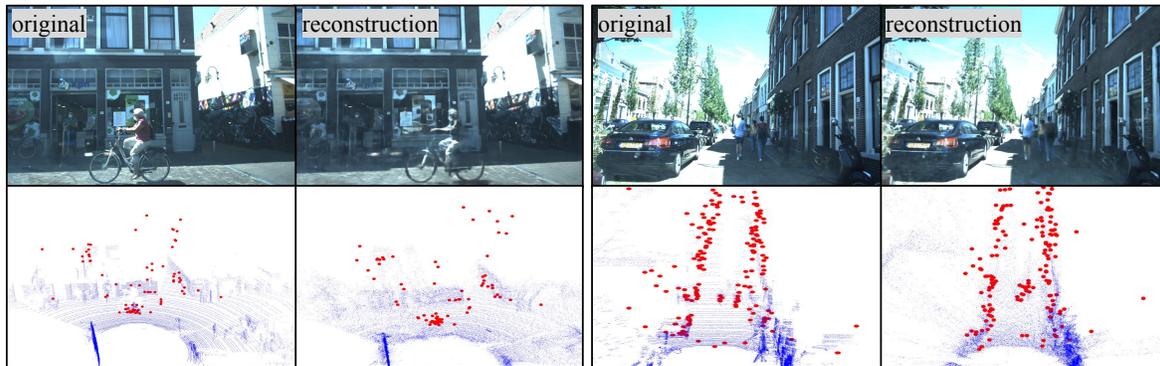


Figure 7. Novel view synthesis results VoD sequences.