



NeuRadar: Neural Radiance Fields for Automotive Radar Point Clouds

Downloaded from: <https://research.chalmers.se>, 2025-09-25 00:29 UTC

Citation for the original published paper (version of record):

Rafidashti, M., Lan, J., Fatemi, M. et al (2025). NeuRadar: Neural Radiance Fields for Automotive Radar Point Clouds. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops: 2488-2498

N.B. When citing this work, cite the original published paper.

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

NeuRadar: Neural Radiance Fields for Automotive Radar Point Clouds

Mahan Rafidashti^{†,1,2} Ji Lan^{†,1,2} Maryam Fatemi¹ Junsheng Fu¹
 Lars Hammarstrand² Lennart Svensson²
¹Zenseact ²Chalmers University of Technology
 {firstname.lastname}@{zenseact.com, chalmers.se}



Figure 1. NeuRadar generates radar point clouds alongside camera and lidar data for novel viewpoints and altered scenes. This figure illustrates images, radar point clouds (in red), and lidar point clouds (in blue) generated by NeuRadar for two sequences in ZOD.

Abstract

Radar is an important sensor for autonomous driving (AD) systems due to its robustness to adverse weather and different lighting conditions. Novel view synthesis using neural radiance fields (NeRFs) has recently received considerable attention in AD due to its potential to enable efficient testing and validation but remains unexplored for radar point clouds. In this paper, we present NeuRadar, a NeRF-based model that jointly generates radar point clouds, camera images, and lidar point clouds. We explore set-based object detection methods such as DETR, and propose an encoder-based solution grounded in the NeRF geometry for improved generalizability. We propose both a deterministic and a probabilistic point cloud representation to accurately model the radar behavior, with the latter being able to capture radar’s stochastic behavior. We achieve realistic reconstruction results for two automotive datasets, establishing a baseline for NeRF-based radar point cloud simulation models. In addition, we release radar data for ZOD’s Sequences and Drives to enable further research in this field. To encourage further development of radar NeRFs, we release the [source code](#) for NeuRadar.

1. Introduction

Neural radiance fields (NeRFs) may become an essential tool for testing and validating autonomous systems [28, 53]. For example, simulating realistic data from critical and

near-collision scenarios allows autonomous driving (AD) companies to test key system aspects efficiently and safely, avoiding the high costs and risks associated with studying these scenarios in real traffic. Developing NeRF models capable of representing general traffic scenarios and the sensor setups used in AD is therefore of vital importance.

Several methods construct NeRF models to generate camera and lidar data [44, 52] for AD. A common approach is to decompose the field into one component for the static background and another for each dynamic object. Recent papers have introduced neural feature fields, a type of NeRF model that outputs a feature vector for each pair of 3D point and view direction. These methods show promising results for camera and lidar data, particularly for poses close to those observed in the training data.

Many AD systems also incorporate radar sensors for increased robustness. Radars offer advantages in cases where optical detection is limited, such as low visibility and adverse weather conditions, and can serve as complementary sensors to lidar and camera in automotive perception systems [5, 40, 41]. Unlike lidar and camera data, radar sensors often provide a sparse point cloud for each measurement scan [20, 55]. Existing NeRF models [5, 21] cannot process such data effectively, and the task of learning a joint model to reconstruct camera, lidar, and radar point clouds remains unexplored.

In this paper, we propose a NeRF model that jointly handles camera, lidar, and radar data for AD. Our solution is based on NeuRAD and uses a joint feature field to represent all three sensors. Given the complexity of modeling

[†]: These authors contributed equally to this work.

radar point clouds as a function of geometry, we take a data-driven approach to learn the mapping from the feature field to radar detections, while grounding the predictions in the NeRF geometry for improved novel view synthesis. Radar detections are known for being noisy and sparse, and even slight differences in sensor and actor poses can yield markedly different radar returns for the same scene. To address this, our model can function in both deterministic and probabilistic modes: it can generate a single predicted point cloud based on the sensor’s pose, or produce a random finite set that reflects the distribution of radar detections across the scene and their return probabilities. We establish a baseline and provide suitable metrics to evaluate the performance of this method using two datasets: View-of-Delft (VoD) [36], and a new extended version of Zenseact Open Dataset (ZOD) [1], which now includes radar point cloud data for the *Sequences* and *Drives* of ZOD. To sum up our contributions:

- We present the first NeRF model that jointly synthesizes radar, camera, and lidar data, addressing a gap in multi-sensor 3D scene representation and enhancing the scope of unified sensor data rendering in AD applications.
- We introduce a data-driven radar model, employing a transformer-based network to create both deterministic and probabilistic representations of radar point clouds, thereby improving adaptability to real-world scenarios.
- To facilitate future exploration of multi-modal NeRFs that include radar data, we release radar point cloud data for ZOD’s *Sequences* and *Drives*, contributing valuable resources to the research community.
- We establish a baseline and evaluation metrics designed for NeRF-based radar point cloud simulation, laying the groundwork for continued research in this area.

2. Related Work

2.1. NeRFs for Automotive Data

The task of building NeRF models for autonomous driving data has recently attracted considerable attention [28, 44, 47, 50–52]. MARS [47] proposes a modular framework that allows users to use various NeRF-based methods for simulating dynamic actors and static backgrounds. However, since it does not natively support lidar data and its performance relies on access to depth maps, MARS’ applicability to certain datasets is limited. UniSim [52], based on Neural Feature Fields (NFF) [31, 32], enables realistic multi-modal sensor simulation for large-scale dynamic driving scenes. NeuRAD [44] extends UniSim by modeling both static and dynamic elements while integrating effective sensor modeling techniques. However, these methods, including recent work like UniCal [54], do not consider radar and thus fail to simulate radar detections. Our work draws inspiration from NeuRAD and expands its capabilities by incorporat-

ing radar data to simulate radar point clouds effectively.

2.2. Radar Simulation

The simulation of radar data can be categorized from different perspectives [29]. Here, we briefly describe model-based radar simulation methods, which involve modeling aspects of the environment and radar operations, and data-driven radar simulation methods, which focus on the detection results on an object or the environment [23, 29].

Model-based Radar Simulation: Ray-tracing methods model the propagation and interaction of radar waves by treating them as rays that travel through a 3D environment, reflecting and scattering upon encountering surfaces [11, 12, 19, 35, 58]. These methods involve creating a detailed explicit representation of the environment and objects, limiting their scalability in complex automotive scenarios, and making them computationally intensive. In contrast, our method also passes rays through the scene, but it relies on the NeRF model to implicitly represent the scene and is purely data-driven.

In object tracking, the radar measurement models represent reflections from multiple points on extended objects, such as vehicles, using probabilistic or geometric models. These models use structures like specific high-reflectance points and plane reflectors to create realistic radar detections [6, 14, 16, 17, 48]. While useful for modeling radar reflections from single objects, these models include many simplifying assumptions and are unsuitable for complex environments including static and dynamic objects.

Data-driven Radar Simulation: Data-driven models offer the flexibility to capture real-world phenomena that traditional physics-based models struggle to emulate. In [2], the authors simulate point clouds using a CNN to estimate the parameters of a Gaussian Mixture Model based on front camera images. However, in this method, radar detection simulation is restricted to the front camera’s field of view.

Recent interest has emerged in using neural rendering to represent various types of radar data. One notable application is in Synthetic Aperture Radar (SAR), where NeRF-inspired models enhance 3D scene reconstruction [22, 24, 27]. Radar Fields [5], a modified NeRF, operates with raw frequency-domain Frequency-Modulated Continuous-Wave (FMCW) radar data, learning implicit scene geometry and reflectance models. By bypassing conventional volume rendering and directly modeling in frequency space, this approach effectively extracts dense 3D occupancy data from 2D range-azimuth maps. Another direction is explored by DART (Doppler-Aided Radar Tomography) [21], which uses radar’s unique capability to measure motion through Doppler effects. DART applies a NeRF-inspired pipeline to synthesize range-Doppler images from new viewpoints, implicitly capturing the tomographic properties of radar signals. Notably, none of the aforementioned methods have

directly used radar point clouds.

2.3. Object Detection

The initial series of object detectors using deep learning addressed object detection indirectly by defining surrogate regression and classification problems on a substantial number of proposals [8, 38], anchors [26], or predefined grids of potential object locations [57]. Alternatively, object detection can be viewed as a direct set prediction task. DETR [9] employs a transformer-based model to infer a fixed number of predictions and uses a set-based global loss that forces unique matching between these predictions and the ground truth through bipartite matching. In [10], the authors introduce a Decoder-Free Fully Transformer-based object detection network. Following these works, we view radar detection as a set prediction problem, integrating an NFF with an encoder-only transformer to extract radar detections.

In applications requiring reliable object detection, such as autonomous driving, capturing uncertainties is crucial for safe and robust decision-making. Probabilistic object detection methods provide probability distributions for each detection, rather than fixed outputs, helping assess detection reliability and reduce overconfident errors [13, 15, 30, 43]. A probabilistic approach to DETR’s set prediction problem is explored in [18], where the predicted set of detections is modeled as a Random Finite Set (RFS). The authors use Negative Log Likelihood as a loss function and metric, which takes into account the uncertainty of the predictions [37]. To acknowledge the stochastic nature of radar detections, we incorporate ideas from [18] into our model, enabling it to model the distribution of radar detections.

3. Preliminaries

As our method builds upon NeuRAD, we briefly describe its essential components. We explain the neural feature field, which is the core of NeuRAD. In addition, we present the modeling of camera and lidar sensors and the construction of corresponding losses. The objective is to present sufficient details to understand our radar model extension.

3.1. Neural Feature Field

At its core, NeuRAD learns a Neural Feature Field (NFF), based on iNGP [34]. The NFF is defined as a continuous and learned function with learnable parameters θ :

$$(s, \mathbf{f}) = f_{\theta}(\mathbf{x}, t, \mathbf{d}), \quad (1)$$

where the inputs are a 3D point $\mathbf{x} \in \mathbb{R}^3$, time $t \in \mathbb{R}$, and a normalized view direction $\mathbf{d} \in \mathbb{R}^3$. These inputs are mapped to an implicit geometry described by the signed distance function (SDF) $s \in \mathbb{R}$ and a feature vector $\mathbf{f} \in \mathbb{R}^{N_f}$.

In an NFF, a ray originating from the sensor center $\mathbf{o}$ with a specific normalized view direction $\mathbf{d}$ is defined as

$$\mathbf{r}(\tau) = \mathbf{o} + \tau \mathbf{d}, \quad (2)$$

where τ represents the distance along the ray from the origin $\mathbf{o}$. To render the feature for the ray at time t , NeuRAD samples N_r 3D points $\mathbf{x}_i = \mathbf{o} + \tau_i \mathbf{d}$ and extract the corresponding feature vector $\mathbf{f}_i$ and implicit geometry s_i using (1). The final ray feature is formed using volume rendering:

$$\mathbf{f}(\mathbf{r}) = \sum_{i=1}^{N_r} \omega_i \mathbf{f}_i, \quad \omega_i = \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (3)$$

where α_i represents the opacity at the point $\mathbf{x}_i$. This opacity is approximated by $\alpha_i = 1/(1 + e^{\beta s_i})$, where β is a learnable parameter. Furthermore, w_i is the product of opacity and the cumulative transmittance to τ_i along the ray.

3.2. Sensor Modeling

NeuRAD models the camera by generating a feature map $\mathbf{F} \in \mathbb{R}^{H_f \times W_f \times N_f}$ by volume rendering a set of $H_f \times W_f$ camera rays, following (3). In accordance with [52], this feature map is converted to an RGB image $\mathbf{I}_{\text{rgb}} \in \mathbb{R}^{H_I \times W_I \times 3}$ using a 2D CNN. In practice, the feature map is generated at a lower spatial resolution $H_f \times W_f$ compared to the final image resolution $H_I \times W_I$, drastically reducing the number of ray queries needed.

The lidar is treated similarly, but instead of rendering an image, NeuRAD renders depth, intensity, and ray drop probability for each lidar ray. In the same way as the camera, a lidar feature is rendered for each ray $\mathbf{r}(\tau) = \mathbf{o} + \tau \mathbf{d}$ of the lidar using (3). However, in contrast to the camera model, intensity and ray drop probability are predicted independently per ray by processing each rendered ray feature through a Multi-Layer Perceptron (MLP). Further, the ray depth, which is the depth at which the ray is reflected off the scene towards the sensor, is modeled directly using the sampled 3D points as the expected depth of the ray

$$D(\mathbf{r}) = \sum_{i=1}^{N_r} w_i \tau_i, \quad (4)$$

where w_i is the same as the term used in (3).

3.3. Image and Lidar Losses

NeuRAD simultaneously optimizes all model components, employing both camera and lidar data for supervision to construct the joint loss:

$$\mathcal{L}^{\text{NeuRAD}} = \mathcal{L}^{\text{image}} + \mathcal{L}^{\text{lidar}}. \quad (5)$$

The image loss $\mathcal{L}^{\text{image}}$ is computed patch-wise and summed across multiple patches, and includes a reconstruction loss $\mathcal{L}^{\text{rgb}}$ and a perceptual loss $\mathcal{L}^{\text{vgg}}$.

For the lidar loss we have four terms, one reconstruction loss for each output, *i.e.*, depth loss $\mathcal{L}^d$ and intensity loss $\mathcal{L}^{\text{int}}$, a binary cross-entropy loss for the ray drop probability $\mathcal{L}^{p_d}$, and an additional term to suppress rendering weights

of distant samples $\mathcal{L}^w$. These terms are considered independently for each lidar ray.

4. Method

We aim to generate radar point clouds from novel view-points using a neural representation of a 3D scene. Since radar point clouds are sparse and noisy, we propose training our model using camera, lidar, and radar data simultaneously. This allows us to leverage the rich semantic and geometric information from camera and lidar data to inform our radar model about the environment. To learn a model of automotive scenes using all three modalities, we build upon NeuRAD and extend it to include radar data. We call this extended model NeuRadar.

4.1. Radar Modeling

Although both automotive lidar and radar generate point clouds via transmitted and reflected electromagnetic waves, their measurement principles and output differ significantly. Radar’s longer wavelengths (millimeter-scale versus lidar’s nanometers) make conductive materials like metal highly visible, while small objects like raindrops remain nearly undetectable. Additionally, AD radars have a larger beam width [4], which causes reflecting waves from close structures to interact, producing complex reflection patterns. Thus, while camera and lidar data can be modeled using local features from narrow beams in NeRF, radar detections can strongly depend on the surroundings, motivating a different modeling approach.

Various approaches to modeling radar detections could be explored. One approach is to mimic the lidar model in NeuRAD and assume that radar detections appear at the expected depth along radar rays (see Eq. (4)). However, this model does not fully account for radar properties, as radar detections can appear far from surfaces. Another approach is to formulate the task as an object detection problem, where a conventional detection network predicts detection positions directly from volume-rendered NFF features. However, this approach gives the decoder complete flexibility and requires the network to predict radar point positions entirely without using the geometry provided by the NFF. In Appendix C.2, we empirically show that a naive version of this approach, which tends to memorize point clouds, is incapable of true novel view synthesis.

We address these deficiencies by combining the strengths of the two discussed approaches. Specifically, we extract the expected depth from the NFF, leveraging NeRF geometry to enhance novel view synthesis. Meanwhile, we use a simple transformer to model the difference between where the simulated radar rays bounce (estimated using (4)) and the positions of radar detections, effectively accounting for radar properties. Our method is illustrated in Fig. 2 and described in detail in the following subsections.

4.1.1. Radar Feature Rendering

Given the position and pose of the radar sensor in space, our objective is to predict its output: a set of points in 3D space. We estimate the spatial coordinates of these points along with confidence measures that indicate the likelihood that each predicted point corresponds to an actual radar reflection. In addition, we quantify the uncertainty in spatial coordinates for each radar point. To this end, we extract relevant information from the NFF—specifically, the expected depth and a feature vector for each point in the scene—and use a decoder network to predict the point cloud coordinates and other parameters.

A key step in this process is constructing a set of rays that traverse the scene, extracting spatial and feature information from the NFF for our radar decoder model. To retrieve scene information, we parameterize these rays as

$$\{\mathbf{r}_{jk}(\tau) : \mathbf{r}_{jk}(\tau) = \mathbf{o} + \tau \mathbf{d}_{jk}\}_{j,k=1}^{N_\varphi, N_\theta}, \quad (6)$$

where $\mathbf{o}$ is the sensor origin and the ray directions $\mathbf{d}_{jk}$ are unit vectors parameterized by azimuth angle φ_j and elevation angle θ_k which are uniformly spaced within the radar field of view. The number of rays in azimuth (N_φ) and elevation (N_θ) is determined by the radar field of view and ray divergence hyperparameters (δ_φ and δ_θ). Fig. 3 illustrates this design, which ensures that radar ray rendering captures all detectable scene information.

For each ray in the set, we compute a feature vector and the expected depth of the reflected ray using volumetric rendering, as described in (3) and (4), respectively. When generating a single ray, the azimuth and elevation information is known. With the supplementary information of the expected depth from the NFF, we determine the ray return positions, consisting of (*azimuth, elevation, depth*). This spherical coordinate is then transformed into Cartesian coordinates and embedded into a feature vector. The final feature vector for a single ray is then obtained by adding the embeddings to the original feature vector obtained from (3). The final feature vector, hence, combines features from the NFF with knowledge about the geometry. As our radar rays share the NFF with the lidar and camera model, the model leverages information about the 3D geometry of the scene obtained from these sensors when rendering its features.

4.1.2. Radar Decoder

The radar decoder takes the extracted features and predicts the radar point cloud. The overall architecture contains a transformer encoder that performs self-attention across the radar ray features, followed by small MLP heads for the different prediction tasks, see Fig. 2 (b).

We formulate one deterministic and one probabilistic version of the decoder. The deterministic method predicts offsets from the known ray return positions. By adding the offsets to the known positions, we obtain the desired set of

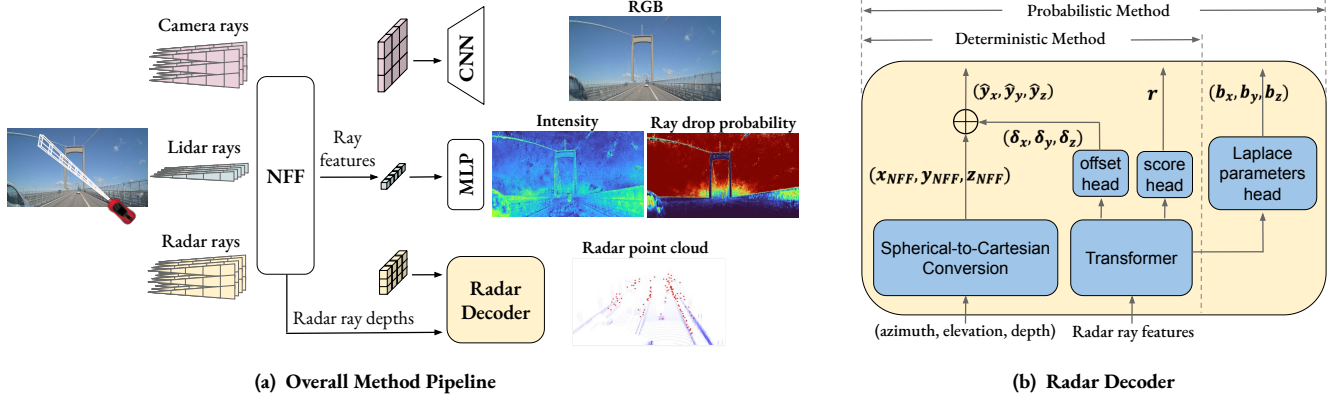


Figure 2. NeuRadAR: our multimodal novel view rendering method for autonomous driving. (a) Rays from each sensor modality render ray features from the NFF. Camera and lidar branches decode their ray features into RGB values via an upsampling CNN and into lidar ray drop probability and intensity via MLPs, respectively. Radar ray features, along with estimated depths from the NFF, generate a radar point cloud via a specialized radar decoder. (b) The ray return position $(x_{NFF}, y_{NFF}, z_{NFF})$ is obtained from the known azimuth, elevation, and estimated depth, while a Transformer predicts offsets $(\delta_x, \delta_y, \delta_z)$ and a detection confidence score r . The sum of the ray return position and offsets determines the final position. As an extension, the probabilistic method uses a Laplace parameter head to predict the scaling parameters. The set $\{r, x_{NFF}, \delta_x, b_x, y_{NFF}, \delta_y, b_y, z_{NFF}, \delta_z, b_z\}$ provides all necessary information for modeling the MB RFS.

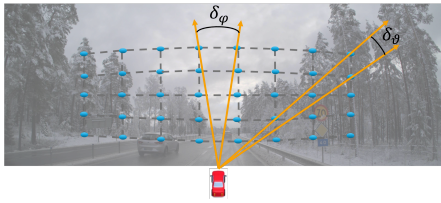


Figure 3. Illustration of our design, where radar-projected rays cover the full sensor field of view (FOV). The virtual grid (grey dashed lines) indicates the complete FOV, with blue points denoting uniformly sampled locations. The direction from the sensor to each blue point defines a ray.

3D points. The decoder also predicts a confidence score for each point representing the probability that it corresponds to an actual detection. By thresholding these scores, the model outputs the estimated radar point cloud. As an extension, the probabilistic method predicts additional parameters to model the distribution of the radar point cloud. In this way, the output is a distribution rather than a fixed set of points, allowing different radar point clouds to be sampled.

4.1.3. Point Cloud Representation

We represent the radar point cloud using both deterministic and probabilistic approaches. In the deterministic approach, for each of the rays, we obtain the Cartesian position of the ray return, $\hat{\mathbf{y}}_{NFF} = (x_{NFF}, y_{NFF}, z_{NFF})^T$, from the NFF. We then predict an offset from these coordinates, $\delta_{\hat{\mathbf{y}}} = (\delta_x, \delta_y, \delta_z)^T$, and a detection confidence score r , using the transformer. The final detection position, $\hat{\mathbf{y}} = (\delta_x, \delta_y, \delta_z)^T$, is computed by $\hat{\mathbf{y}} = \hat{\mathbf{y}}_{NFF} + \delta_{\hat{\mathbf{y}}}$.

With this approach, the model directly renders a point

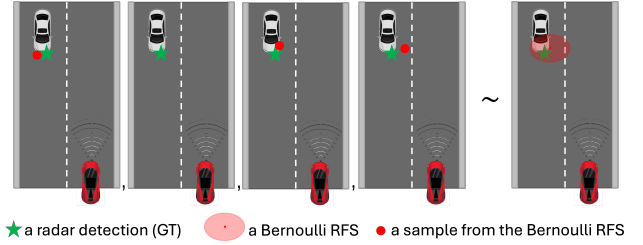


Figure 4. Four sampled sets (left) drawn from a Bernoulli RFS (right) with an existence probability of $r = 0.75$. The RFS accounts for both the absence of objects and spatial uncertainties. The image is provided for context only.

cloud that is deterministic in the sense that, for a given input, the positions of the detections are fixed. From a simulation perspective, this could be desirable, *e.g.*, to ensure repeatability when testing downstream tasks.

We also provide a probabilistic representation of the radar point cloud to capture uncertainties in radar detections and to allow for rendering slightly different radar point clouds for a given input. Inspired by [18], we model the probabilistic point cloud as a Multi-Bernoulli Random Finite Set (MB-RFS),

$$\mathbb{Y} = \bigcup_{i=1}^{N_{\text{rays}}} \mathbb{Y}_i, \quad (7)$$

where $\mathbb{Y}_1, \dots, \mathbb{Y}_{N_{\text{rays}}}$ are Bernoulli RFSs describing possible individual radar detections. The radar point cloud is, thus, modeled as a set of detections for which the cardinality and the points themselves are random variables.

Each Bernoulli RFS $\mathbb{Y}_i$ is defined by

$$f_{B_i}(\mathbb{Y}_i) = \begin{cases} 1 - r_i, & \text{if } \mathbb{Y}_i = \emptyset, \\ r_i p_i(\mathbf{y}), & \text{if } \mathbb{Y}_i = \{\mathbf{y}\}, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where r_i is the existence probability and the density $p_i(\mathbf{y})$ describes the position of the radar detection. In practice, the existence probability r_i is the same as the detection confidence score defined in the deterministic method. We also assume that the density $p_i(\mathbf{y})$ can be factorized as

$$p_i(\mathbf{y}) = p_i(x)p_i(y)p_i(z), \quad (9)$$

where the individual densities are modeled as Laplacian distributions parameterized by (μ_{x_i}, b_{x_i}) , (μ_{y_i}, b_{y_i}) and (μ_{z_i}, b_{z_i}) , respectively. In practice, $(\mu_{x_i}, \mu_{y_i}, \mu_{z_i})$ is the same as the detection positions $(\delta_x, \delta_y, \delta_z)$ defined in the deterministic method. Thus, for each ray i , we predict the set of parameters $\{r_i, x_{\text{NFF}_i}, \delta_{x_i}, b_{x_i}, y_{\text{NFF}_i}, \delta_{y_i}, b_{y_i}, z_{\text{NFF}_i}, \delta_{z_i}, b_{z_i}\}$ that describe the corresponding Bernoulli distribution.

Using (7) and (8) we can express the set probability distribution for the complete radar point cloud as a multi-Bernoulli (MB) distribution, defined as

$$f_{\text{MB}}(\mathbb{Y}) = \sum_{\cup_{i \in \mathbb{I}} \mathbb{Y}_i = \mathbb{Y}} \prod_{i=1}^{N_q} f_{B_i}(\mathbb{Y}_i), \quad (10)$$

where $\cup_{i \in \mathbb{I}} \mathbb{Y}_i = \mathbb{Y}$ represents the sum over all disjoint sets with union $\mathbb{Y}$. Note that each Bernoulli in the MB is independent making it convenient to generate a radar point cloud from (10) by generating samples from each Bernoulli separately using (8). An example of a Bernoulli RFS prediction and corresponding samples is shown in Fig. 4.

4.2. Losses

Our total loss is formed by extending the NeuRAD loss defined in (5) with a radar loss $\mathcal{L}^{\text{radar}}$,

$$\mathcal{L} = \mathcal{L}^{\text{NeuRAD}} + \mathcal{L}^{\text{radar}}. \quad (11)$$

As previously mentioned, we consider two types of radar models, one deterministic and one probabilistic, each having its own formulation of $\mathcal{L}^{\text{radar}}$. In this section, we present both of these losses and how we match ground truth radar detections to model predictions to calculate the loss.

Matching: Similar to [9, 18], we find a matching or an assignment between predictions and ground truth by solving an optimal assignment problem. Let $\mathbb{Y} = \{\mathbf{y}_j\}_{j=1}^{|\mathbb{Y}|}$ denote the ground truth set of radar points, and $\hat{\mathbb{Y}} = \{\hat{\mathbf{y}}_i\}_{i=1}^{|\hat{\mathbb{Y}}|}$ the set of radar point predictions; for the probabilistic model, we set $\hat{\mathbf{y}}_i = (\mu_{x_i}, \mu_{y_i}, \mu_{z_i})^T$. Note that r_i represents the

probability that $\hat{\mathbf{y}}_i$ is a reflected radar point. We ensure that $|\hat{\mathbb{Y}}| > |\mathbb{Y}|$ by setting the number of rays such that it is larger than the maximum number of radar detections across all scans in the dataset. The assignment problem can be formulated as

$$\min_{\mathbf{A}} \sum_i \sum_j \mathbf{C}_{i,j} \mathbf{A}_{i,j}, \quad (12a)$$

$$\text{s.t.} \quad \sum_{i=1}^{|\hat{\mathbb{Y}}|} \mathbf{A}_{i,j} = 1, \quad \sum_{j=1}^{|\mathbb{Y}|} \mathbf{A}_{i,j} \leq 1, \quad (12b)$$

$$\mathbf{C}_{i,j} = \|\hat{\mathbf{y}}_i - \mathbf{y}_j\| - \log(r_i), \quad (12c)$$

where $\mathbf{C}$ is a cost matrix and $\mathbf{A}$ is the assignment matrix. In (12c), both the Euclidean distance between $\hat{\mathbf{y}}_k$ and $\mathbf{y}_l$, and the existence probability of the k th prediction, are considered jointly. Constrained by (12b), all ground truth objects will be assigned to a prediction each, while some predictions will remain unassigned. By optimizing (12a), we obtain the optimal assignment, denoted by $\mathbf{A}^*$, and the set $\gamma = \{i \in \mathbb{N}_{|\hat{\mathbb{Y}}|}, j \in \mathbb{N}_{|\mathbb{Y}|} \mid \forall i, j : \mathbf{A}_{i,j}^* = 1\}$, which records all matched pairs.

Deterministic Radar Loss: For the deterministic radar model, the loss is defined as

$$\mathcal{L}_{\text{det}}^{\text{radar}} = \sum_{(i,j) \in \gamma} (\|\hat{\mathbf{y}}_i - \mathbf{y}_j\| - \log(r_i)) - \sum_{(i,j) \notin \gamma} \log(1 - r_i). \quad (13)$$

In (13), the first term minimizes the existence probability of the unassigned predictions. Additionally, the second term reduces the Euclidean distance between matched pairs of predicted and ground truth radar points and maximizes the existence probability of the assigned predictions.

Probabilistic Radar Loss: In the probabilistic method, we approximate the negative log-likelihood of the MB distribution $-\log f_{\text{MB}}(\mathbb{Y})$ in (10) by the term corresponding to $\mathbf{A}^*$ and use that as our loss:

$$\begin{aligned} \mathcal{L}_{\text{prob}}^{\text{radar}} = & - \sum_{(i,j) \notin \gamma} \log(1 - r_i) \\ & - \sum_{(i,j) \in \gamma} \log(r_i p_{i,x}(\mathbf{y}_{j,x}) p_{i,y}(\mathbf{y}_{j,y}) p_{i,z}(\mathbf{y}_{j,z})). \end{aligned} \quad (14)$$

Similar to $\mathcal{L}_{\text{det}}^{\text{radar}}$, $\mathcal{L}_{\text{prob}}^{\text{radar}}$ minimizes the existence probability of the unassigned predictions and maximizes the existence probability of the assigned predictions. In contrast, the second term in (14) maximizes the probability that a matched radar point prediction accurately represents its corresponding ground truth radar point. The similarity between $\mathcal{L}_{\text{det}}^{\text{radar}}$ and $\mathcal{L}_{\text{prob}}^{\text{radar}}$ becomes even more striking by noting that $\log p_{i,x}(\mathbf{y}_{j,x}) = |\mathbf{y}_{j,x} - \mu_{x_i}| / (2b_{x_i}) + \log(2b_{x_i})$.

5. Datasets

In this section we provide an overview of available datasets suitable for studying automotive radar point cloud generation based on neural representations of the scene. We also introduce a new dataset by extending the existing ZOD [1] with radar point cloud data.

5.1. Existing Datasets

To facilitate studies like the one presented in this paper, researchers need datasets that include driving sequences with good-quality camera images, lidar, and radar point clouds, along with a permissive license for broad accessibility. Among existing AD datasets, Pandaset [49], Argoverse [46] and Kitti-360 [25] offer good-quality camera and lidar data with permissive licensing but lack radar point cloud data. Both nuScenes [7] and VoD [36] provide camera images, lidar and radar point clouds with relatively permissive licenses. However, nuScenes radar point cloud is sparse, making it less suitable for the task at hand. Despite VoD’s limitations in image quality, we selected it as one of the datasets used in this paper due to its dense radar point clouds. Nonetheless, we emphasize the need for automotive datasets that better meet essential requirements, including high-quality camera, lidar, and radar data, along with a permissive license to advance research.

5.2. Zenseact Open Dataset

ZOD [1] is a diverse and large-scale multimodal dataset for autonomous driving collected in several European countries which includes a million curated *Frames*, 1473 *Sequences* of 20-second length and 29 extended *Drives* spanning several minutes. Like many contemporary datasets, ZOD includes high-resolution lidar and camera data but was initially provided without radar data.

In this paper, we extend ZOD *Sequences* and *Drives* by adding radar point cloud data. Radar point clouds are obtained from a Continental Radar ARS513 B1 sample – year 2022, and are captured at every 60 ms. The addition of radar data together with ZOD’s permissive license enables diverse research applications in automotive perception, such as multimodal scene reconstruction and novel view synthesis, that *includes* radar point cloud data. Further details on the radar data are in the supplementary material.

6. Experiments

In this section, we evaluate NeuRadar’s ability to reconstruct radar point clouds alongside camera and lidar data for VoD and ZOD datasets. We evaluate the performance of the model in terms of reconstruction accuracy, and extrapolation to confirm the consistency and generalization of NeuRadar in various scenarios. In this work, we use Chamfer Distance (CD) [3] and Earth Mover’s Distance (EMD)

[39] to measure the similarity between the generated point cloud and the ground truth point cloud.

As a baseline, we adapt NeuRAD’s lidar point cloud generation method to generate radar detections. For each radar ray in (6), we predict the depth according to (4), which together with ray direction gives us possible radar detection $\hat{y}$. The probability of generating this detection is modelled using a simple network fed with the ray features (3) as input. Following the Lidar ray drop probability and intensity prediction methods, we use an MLP. The deterministic loss described in (13) is then used to train this baseline model. The predicted existence probabilities are then used to decide which rays would reflect and result in radar detections. In the baseline method and the deterministic method, we choose detections with existence probability higher than a threshold, here 0.5. In the probabilistic method, we simply sample from the MB RFS.

6.1. Implementation Details

NeuRadar is implemented in nerfstudio [42] and is based on the open-source NeuRAD code. For the radar decoder, the learning rate varies from 10^{-3} to 10^{-7} over the first 10,000 iterations, with a warmup period of 5,000 iterations. For the rest of the network, we use the same training schedule as NeuRAD. We train our model for 20,000 iterations, which takes on average three hours on a single NVIDIA A100.

6.2. Novel View Synthesis

To evaluate the novel view synthesis performance of NeuRadar, during training, we hold out every second frame in the sequence and test on this data. For each dataset, we train models on 10 sequences (described in Appendix B) and report the median value of each metric over these sequences. Alongside the metrics for the entire point cloud, we report metrics for radar points within the 30 and 80-meter range of the sensor. Table 1 shows that both of our proposed methods outperform the baseline for both datasets, especially in higher range. The baseline method performs reasonably well near the sensor but degrades as the distance increases. This is demonstrated in Fig. 5, which shows the radar and lidar point clouds rendered alongside the image using the three methods. Furthermore, the probabilistic NeuRadar model performs better than the deterministic model and the baseline for both VoD and ZOD.

6.3. Novel Scenario Generation

In this section, we explore NeuRadar’s capability to synthesize radar point clouds from entirely novel viewpoints, from which the radar has not previously observed the scene. There is no ground truth available for these viewpoints, therefore, we present qualitative results through visualizations to present NeuRadar’s ability to generalize beyond observed viewpoints. Figure 1 illustrates the results of re-

Table 1. Novel view synthesis performance comparison of the baseline model, our deterministic model (Det.), and our probabilistic model (Prob.). We evaluated the results for predictions and ground truth within 30 and 80 meters of the radar sensor (indicated by the subscript) and the entire point cloud (no subscript). Lower CD and EMD values indicate better performance.

Method	ZOD						VoD					
	CD ₃₀	EMD ₃₀	CD ₈₀	EMD ₈₀	CD	EMD	CD ₃₀	EMD ₃₀	CD ₈₀	EMD ₈₀	CD	EMD
Baseline	2.854	2.981	4.257	4.460	5.536	5.851	2.965	2.799	3.924	4.067	4.698	4.966
Det.	2.592	2.647	3.981	4.050	4.439	5.136	2.122	2.405	3.606	3.256	3.785	3.890
Prob.	2.309	2.601	3.846	4.025	4.298	5.348	2.091	2.272	3.239	3.189	3.388	3.734

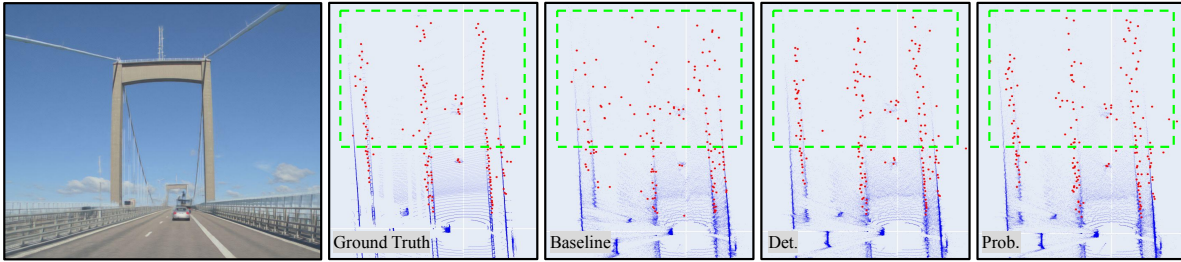


Figure 5. Novel view synthesis for interpolated frames including radar (red) and lidar (blue) detections. The Deterministic (Det.) and Probabilistic (Prob.) methods outperform the baseline at larger distances.

moving dynamic actors from the scene and shifting the ego pose 3 meters laterally to simulate lane change. The visualizations indicate that NeuRadar successfully generates a realistic radar point cloud, reinforcing its potential for novel scenario generation in autonomous driving applications.

6.4. Ablations

We present experiments on two key model design choices, conducted on our probabilistic radar model across three sequences in two datasets. The first experiment investigates two popular choices for the spatial density used in the Bernoulli distribution: Laplace and Gaussian. As shown in Tab. 2, models using the Laplace distribution outperform those using the Gaussian distribution. Thus, we advocate using the Laplace distribution in our probabilistic radar model.

Table 2. Results for two spatial density functions used in the MB.

Dataset	Density	CD ↓	EMD ↓
ZOD	Laplace	3.92	4.94
	Gaussian	4.91	5.86
VoD	Laplace	4.02	4.28
	Gaussian	4.28	4.44

The second experiment evaluates three designs in the radar decoder. As shown in Tab. 3, models employing a Transformer-based architecture outperform those using either an MLP or a CNN-based architecture. This is likely because the Transformer’s self-attention mechanism enables

the radar decoder to capture long-range dependencies and contextual information in radar data more effectively.

Table 3. Results for different models used in the radar decoder.

Model	ZOD		VoD	
	CD ↓	EMD ↓	CD ↓	EMD ↓
MLP	4.93	6.14	4.30	4.61
CNN	5.68	6.29	4.51	4.97
Transformer	3.92	4.94	4.02	4.28

7. Conclusion

In this paper, we introduced NeuRadar, a multi-modal NeRF-based simulation method that includes a data-driven radar model. NeuRadar leverages a unified neural feature field to effectively synthesize camera, lidar, and radar data and is capable of generating realistic radar point clouds from novel views and for different scene configurations. To address the challenge of noisy and sparse radar detections, we developed a probabilistic version of NeuRadar that incorporates the multi-Bernoulli distribution. This version produces a random finite set reflecting the distribution of radar detections across the scene and their return probabilities. Additionally, we enhanced the ZOD dataset by adding radar data to its *Sequences* and *Drives*. We established a baseline and evaluation metrics for NeRF-based radar point cloud simulation and conducted evaluations on three public AD datasets. Lastly, we release our code to foster further research into automotive radar data simulation.

Acknowledgment

This work was supported by Vinnova, Sweden's Innovation Agency. Computational resources were provided by NAISS at NSC Berzelius, partially funded by the Swedish Research Council, grant agreement no. 2022-06725.

References

- [1] Mina Alibeigi, William Ljungbergh, Adam Tonderski, Georg Hess, Adam Lilja, Carl Lindström, Daria Motorniuk, Junsheng Fu, Jenny Widahl, and Christoffer Petersson. Zenseact open dataset: A large-scale and diverse multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20178–20188, 2023. 2, 7
- [2] Tunc Alkanat and Ashish Pandharipande. Automotive radar point cloud parametric density estimation using camera images. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8636–8640, 2024. 2
- [3] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings: Image Understanding Workshop*, pages 21–27. Science Applications, Inc, 1977. 7
- [4] Igal Bilik. Comparative analysis of radar and lidar technologies for automotive applications. *IEEE Intelligent Transportation Systems Magazine*, 15(1):244–269, 2023. 4
- [5] David Borts, Erich Liang, Tim Brödermann, Andrea Ramazzina, Stefanie Walz, Edoardo Palladin, Jipeng Sun, David Bruggemann, Christos Sakaridis, Luc Van Gool, Mario Bijelic, and Felix Heide. Radar fields: Frequency-space neural scene representations for fmcw radar. In *Proceedings of Special Interest Group on Computer Graphics and Interactive Techniques Conference*, page TBD, 2024. 1, 2
- [6] M. Bühren and Bin Yang. Simulation of automotive radar target lists using a novel approach of object representation. In *2006 IEEE Intelligent Vehicles Symposium*, pages 314–319, 2006. 2
- [7] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yuxin Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2019. 7
- [8] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019. 3
- [9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, page 213–229, Berlin, Heidelberg, 2020. Springer-Verlag. 3, 6
- [10] Peixian Chen, Mengdan Zhang, Yunhang Shen, Kekai Sheng, Yuting Gao, Xing Sun, Ke Li, and Chunhua Shen. Efficient decoder-free object detection with transformers. *ArXiv*, abs/2206.06829, 2022. 3
- [11] Ushemadzoro Chipengo. High fidelity physics-based simulation of a 512-channel 4d-radar sensor for automotive applications. *IEEE Access*, 11:15242–15251, 2023. 2
- [12] Javad Ebrahimizadeh, Alireza Madannejad, Xuesong Cai, Evgenii Vinogradov, and Guy A. E. Vandenbosch. Rcs-based 3-d millimeter-wave channel modeling using quasi-deterministic ray tracing. *IEEE Transactions on Antennas and Propagation*, 72(4):3596–3606, 2024. 2
- [13] Di Feng, Ali Harakeh, Steven L. Waslander, and Klaus Dietmayer. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):9961–9980, 2022. 3
- [14] Karl Granström, Marcus Baum, and Stephan Reuter. Extended object tracking: Introduction, overview, and applications. *Journal of Advances in Information Fusion*, 12, 2017. 2
- [15] David Hall, Feras Dayoub, John Skinner, Haoyang Zhang, Dimity Miller, Peter Corke, Gustavo Carneiro, Anelia Angelova, and Niko Suenderhauf. Probabilistic object detection: Definition and evaluation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020. 3
- [16] Lars Hammarstrand, Malin Lundgren, and Lennart Svensson. Adaptive radar sensor model for tracking structured extended objects. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):1975–1995, 2012. 2
- [17] Lars Hammarstrand, Lennart Svensson, Fredrik Sandblom, and Joakim Sorstedt. Extended object tracking using a radar resolution model. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2371–2386, 2012. 2
- [18] Georg Hess, Christoffer Petersson, and Lennart Svensson. *Object Detection as Probabilistic Set Prediction*, page 550–566. Springer Nature Switzerland, 2022. 3, 5, 6
- [19] Nils Hirsenkorn, Paul Subkowski, Timo Hanke, Alexander Schaermann, Andreas Rauch, Ralph Rasshofer, and Erwin Biebl. A ray launching approach for modeling an fmcw radar system. In *2017 18th International Radar Symposium (IRS)*, pages 1–10, 2017. 2
- [20] Martin Holder, Philipp Rosenberger, Hermann Winner, Thomas D'hondt, Vamsi Prakash Makkapati, Michael Maier, Helmut Schreiber, Zoltan Magosi, Zora Slavik, Oliver Bringmann, and Wolfgang Rosenstiel. Measurements revealing challenges in radar sensor modeling for virtual validation of autonomous driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2616–2622, 2018. 1
- [21] Tianshu Huang, John Miller, Akarsh Prabhakara, Tao Jin, Tarana Laroia, Zico Kolter, and Anthony Rowe. Dart: Implicit doppler tomography for radar novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page TBD, 2024. 1, 2

- [22] J. R. Jamora, Dylan Green, Ander Talley, and Thomas Curry. Utilizing sar imagery in three-dimensional neural radiance fields-based applications. In *Proceedings of SPIE - Algorithms for Synthetic Aperture Radar Imagery XXX*, page 1252002, 2023. 2
- [23] Michał Jasiński. A generic validation scheme for real-time capable automotive radar sensor models integrated into an autonomous driving simulator. In *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 612–617, 2019. 2
- [24] Zhengxin Lei, Feng Xu, Jiangtao Wei, Feng Cai, Feng Wang, and Ya-Qiu Jin. Sar-nerf: Neural radiance fields for synthetic aperture radar multiview representation. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024. 2
- [25] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:3292–3310, 2021. 7
- [26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 3
- [27] Afei Liu, Shuanghui Zhang, Chi Zhang, Shuaifeng Zhi, and Xiang Li. Ranerf: Neural 3-d reconstruction of space targets from isar image sequences. *IEEE Transactions on Geoscience and Remote Sensing*, 61, 2023. 2
- [28] William Ljungbergh, Adam Tonderski, Joakim Johnander, Holger Caesar, Kalle Åström, Michael Felsberg, and Christoffer Petersson. Neuroncap: Photorealistic closed-loop safety testing for autonomous driving. In *Computer Vision – ECCV 2024*, pages 161–177, Cham, 2025. Springer Nature Switzerland. 1, 2
- [29] Zoltan Ferenc Magosi, Hexuan Li, Philipp Rosenberger, Li Wan, and Arno Eichberger. A survey on modelling of automotive radar sensors for virtual test and validation of automated driving. *Sensors*, 22(15):5693, 2022. 2
- [30] Rowan McAllister, Yarin Gal, Alex Kendall, Mark van der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4745–4753, 2017. 3
- [31] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2
- [32] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2
- [33] Thomas Müller. tiny-cuda-nn, 2021. 1
- [34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics.*, 41(4):102:1–102:15, 2022. 3
- [35] Anthony Ngo, Max Paul Bauer, and Michael Resch. A multi-layered approach for measuring the simulation-to-reality gap of radar perception for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 4008–4014, 2021. 2
- [36] Andras Palffy, Ewoud Pool, Srimannarayana Baratam, Julian F. P. Kooij, and Darius M. Gavrilă. Multi-class road user detection with 3+1d radar in the view-of-delft dataset. *IEEE Robotics and Automation Letters*, 7(2):4961–4968, 2022. 2, 7
- [37] Juliano Pinto, Yuxuan Xia, Lennart Svensson, and Henk Wymeersch. An uncertainty-aware performance measure for multi-object tracking. *IEEE Signal Processing Letters*, 28:1689–1693, 2021. 3
- [38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 3
- [39] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 2000. 7
- [40] Nicolas Scheiner, Florian Kraus, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. Object detection for automotive radar point clouds – a comparison. *AI Perspectives*, 3, 2021. 1
- [41] Shunqiao Sun, Athina P. Petropulu, and H. Vincent Poor. Mimo radar for advanced driver-assistance systems and autonomous driving: Advantages and challenges. *IEEE Signal Processing Magazine*, 37(4):98–117, 2020. 1
- [42] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM Special Interest Group on Computer Graphics and Interactive Techniques 2023 Conference Proceedings*, 2023. 7
- [43] Alexander Timans, Christoph-Nikolas Straehle, Kaspar Sakmann, and Eric Nalisnick. Adaptive bounding box uncertainties via two-step conformal prediction. In *Proceedings of the European Conference on Computer Vision*, 2024. 3
- [44] Adam Tonderski, Carl Lindstrom, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. NeuRAD: Neural Rendering for Autonomous Driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14895–14904, Los Alamitos, CA, USA, 2024. IEEE Computer Society. 1, 2
- [45] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 3
- [46] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. 7

- [47] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, Yuxin Huang, Xiaoyu Ye, Zike Yan, Yongliang Shi, Yiyi Liao, and Hao Zhao. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *Artificial Intelligence*, pages 3–15, Singapore, 2024. Springer Nature Singapore. [2](#)
- [48] Yuxuan Xia, Pu Wang, Karl Berntorp, Lennart Svensson, Karl Granström, Hassan Mansour, Petros Boufounos, and Philip V. Orlik. Learning-based extended object tracking using hierarchical truncation measurement model with automotive radar. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):1013–1029, 2021. [2](#)
- [49] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, Yunlong Wang, and Diange Yang. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101, 2021. [7](#)
- [50] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-NeRF: Neural radiance fields for street views. In *International Conference on Learning Representations*, 2023. [2](#)
- [51] Honghui Yang, Sha Zhang, Di Huang, Xiaoyang Wu, Haoyi Zhu, Tong He, Shixiang Tang, Hengshuang Zhao, Qibo Qiu, Binbin Lin, et al. Unipad: A universal pre-training paradigm for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15238–15250, 2024.
- [52] Ze Yang, Yun Chen, Jingkan Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1389–1399, 2023. [1](#), [2](#), [3](#)
- [53] Ze Yang, Sivabalan Manivasagam, Yun Chen, Jingkan Wang, Rui Hu, and Raquel Urtasun. Reconstructing objects in-the-wild for realistic sensor simulation. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11661–11668, 2023. [1](#)
- [54] Ze Yang, George Chen, Haowei Zhang, Kevin Ta, Ioan Andrei Bârsan, Daniel Murphy, Sivabalan Manivasagam, and Raquel Urtasun. Unical: Unified neural sensor calibration. In *European Conference on Computer Vision*, pages 327–345. Springer, 2025. [2](#)
- [55] Matthias Zeller, Jens Behley, Michael Heidingsfeld, and Cyrill Stachniss. Gaussian radar transformer for semantic segmentation in noisy radar data. *IEEE Robotics and Automation Letters*, 8(1):344–351, 2023. [1](#)
- [56] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [3](#)
- [57] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. [3](#)
- [58] Ming Zong, Zhanyu Zhu, and Huaisheng Wang. A simulation method for millimeter-wave radar sensing in traffic intersection based on bidirectional analytical ray-tracing algorithm. *IEEE Sensors Journal*, 23(13):14276–14284, 2023. [2](#)