Radar Fields:
Frequency-Space Neural Scene Representations for FMCW Radar
- Supplementary Information -

David Borts
dborts@princeton.edu
Princeton University

Erich Liang
erliang@princeton.edu
Princeton University

Tim Brödermann
tim.broedermann@vision.ee.ethz.ch
ETH Zürich

Andrea Ramazzina
andrea.ramazzina@daimler.com
Mercedes Benz

Stefanie Walz
sefanie.walz@daimler.com
Mercedes Benz

Edoardo Palladin
edoardo.palladin@torc.ai
Torc Robotics

Jipeng Sun
js2694@princeton.edu
Princeton University

David Bruggemann
brdavid@vision.ee.ethz.ch
ETH Zürich

Christos Sakaridis
csakarid@vision.ee.ethz.ch
ETH Zürich

Luc Van Gool
vangool@vision.ee.ethz.ch
ETH Zürich

Mario Bijelic
mario.bijelic@princeton.edu
Princeton University, Torc Robotics

Felix Heide
fheide@princeton.edu
Princeton University, Torc Robotics

This document provides further detail and additional results to support the findings of the main manuscript. We give detailed descriptions of data preprocessing and occupancy estimation, model architecture and training, 3D reconstruction, evaluation metrics, and dataset acquisition.

Contents

1 ADDITIONAL DETAILS ON METHOD, PREPROCESSING AND TRAINING

In this section, we specify key implementation details of our algorithm, including data preprocessing, model architecture, and training.

1.1 Data Preprocessing

Each sequence in our dataset consists of multiple data streams - GNSS, radar, LiDAR, and RGB camera data. Observations in these modalities are not necessarily synced, so we match the data points from each data stream to its temporally closest radar data point. Each radar data point accepts only one data point match per modality, so data points too far away from any radar observation are discarded. This way, we have corresponding GNSS, LiDAR, and RGB data for any radar frame that we provide as input to our model.

To fit any given scene within a finite bounding box, we rescale the 3D scene uniformly across all dimensions so that it fits within a unit bounding box with corners of $(0, 0, 0)$ and $(1, 1, 1)$. We rescale all scene bounding box axes uniformly to preserve relative scale, with the global scale factor set by the largest axis.

For each ground truth radar FFT frame we train on, we omit the inner-most 75 range-azimuth bins from training supervision (approximately 3 meters around the sensor). This is because, in our collected radar frames, there is a persistent intense reading in these inner bins due to the radar detecting the metallic roof of our data collection vehicle. Because this high intensity circle does not correspond to meaningful scene information and is not visible in other frames, we must ignore the data within these bins to avoid potential artifacts.

Finally, we apply a single global noise floor to all frames in the sequence before training to remove excessive sensor noise from the supervision signal, especially at regions corresponding to empty space in data.

SIGGRAPH Conference Papers '24, July 27–August 01, 2024, Denver, CO, USA

1.2 Computational Occupancy Estimator

During data preprocessing, we also iterate over each ground truth FFT frame in a given input sequence, estimating a probability of occupancy for each range-azimuth bin in that frame. These per-frame probability distributions are stored alongside the processed ground truth frames, and used as an additional supervision signal for our neural occupancy field at train time. To do this, we design a computational occupancy estimator $O(P_r)$ to extract the presence of objects from raw FFT data.

First, we filter out specular sensor saturation artifacts and other noise by computing a dynamic, per-bin noise threshold for each bin in the FFT signal $P_r(\phi, b)$. Note that this differs from how we preprocess the ground truth FFT signal, where we instead apply a lower global noise threshold, as we do not wish to remove reflectance-dependent artifacts from the FFT. We first estimate the noise threshold per azimuth direction ϕ, n_ϕ and per range bin b, n_b independently, such that:

$$n_\phi = \text{median}\left(P_r(\phi, b)\right) \text{ (over all bins } b \text{ with fixed } \phi) \tag{1}$$

$$n_b = \text{median}\left(P_r(\phi, b)\right) \text{ (over all azimuth angles } \phi \text{ with fixed } b) \tag{2}$$

We then compare these per-azimuth and per-range thresholds pairwise to get a grid of unique thresholds for each azimuth-range bin. The final noise threshold T is

$$T(\phi, b) = 2 * \max\left(n_\phi, n_b\right). \tag{3}$$

This allows us to account for range-dependent noise and specular highlights. Specular highlights occur across ranges for fixed azimuth angles, increasing the received median power per send beam as a scene reflector saturates the sensor. Received signal intensity is also range-dependent due to the inverse square law, as illustrated in Eq. 7 of the main manuscript, meaning that a noise estimate per distance is useful for estimating occupancy. We apply our dynamic threshold T,

$$P'_r(\phi, b) = \begin{cases} P_r & \text{if } P_r(\phi, b) \geq T(\phi, b) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where P'_r is the thresholded measurement. Next, we iterate over each range-azimuth bin of the thresholded FFT P'_r and estimate a probability of occupancy for that bin using a simple Bayesian update rule and occlusion model, building on the algorithm proposed in [Werber et al. 2015]. In particular, we compute a probability of occupancy p_r for each FFT bin,

$$p_r(\phi, b) = P'_r(\phi, b) \exp\left(\delta(P'_r(\phi, b) - P_o)\right), \tag{5}$$

assuming that the received power is normalized such that $P'_r \in [0, 1]$, with δ and P_o being hyperparameters governing the sensitivity of the estimate. We then combine the above equation with an occlusion model, such that the occupancy probability at a given range also depends on the signal intensity at closer ranges. In other words, an FFT bin with a large signal "casts a shadow" of occupancy on bins behind it that attenuates over distance, thereby modeling occlusion and filling out gaps in the measurement with,

$$p_n(\phi, b) = \max\left(p_r(\phi, b),\ P_r(\phi, b_p) \exp\left(\frac{\Delta x}{\Delta b}\right)\right), \tag{6}$$

$$\text{with} \quad \Delta x = b_p - b, \tag{7}$$

and b_p being the bin at which the last strong reflection from an object was detected, such that $b_p < b$. We decay the signal over distance Δb. Finally, we use p_r to perform a Bayesian update of our occupancy probabilities. Our occupancy probabilities $O(P_r)$ are all initialized to 0.5 and updated as

$$O(P_r) = \frac{p_r p_c}{p_r p_c + (1 - p_r)(1 - p_c)}, \tag{8}$$

with p_c being the previous probability estimate and p_r being the new estimate computed using our above equations.

Finally the predicted $O(P_r)$ allows us to directly supervise the predicted occupancy of our model. Note that, during training, for each sampled bin in a specific frame, we only supervise predicted occupancy with the computed $O(P_r)$ from that same frame, avoiding artifacts from resampling a spherical radar representation from a Cartesian global occupancy map. In this sense, $O(P_r)$ is local, computed per-frame without input from any other frames, and supervises the reconstruction for one frame only. It is fundamentally different from our 3D learned occupancy field, which synthesizes information across frames into a unified representation.

A visualization of $O(P_r)$ is shown in Fig. 2, while the benefit of using $O(P_r)$ as an additional supervision signal is shown in Fig. 1.

1.3 Position and Azimuth Encoding

Before querying our neural networks with any position or view angle information, we apply input encodings to improve training convergence. For 3D positions (x, y, z), we apply a multiresolution hash grid encoding [Müller et al. 2022] with final grid resolution of $512 \times 512 \times 512$. We use 16 layers, with the per-level scale computed as $\exp(\frac{log_2(r/16)}{15})$, where r is the resolution. For view angle, we encode it via spherical harmonics with 4 frequency bands.

1.4 Sampling

During each training iteration, we use a batch size of 16 randomly sampled radar frames from any given input sequence. From each frame in a batch, we sample 200 random azimuth angles and 900 bins per-azimuth from that frame. For each sampled azimuth, we super-sample 9 additional rays within the elliptical cone of sensor beam divergence to perform physics-based importance sampling.

1.5 Model Architecture and Training Details

We utilize TCNN [Müller et al. 2022] as the backbone for our neural networks and input encodings to accelerate optimization. We use the ADAMW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We also use a learning rate schedule, with an initial learning rate of 0.001 that decreases exponentially such that it reaches 0.0001 by the final training iteration. For our experiments, training is done over 500 iterations on a single NVIDIA A100.

1.6 3D Voxel Grid Reconstruction

To generate voxel grids from our model, we query our occupancy field with an evenly-spaced grid of 360 by 360 by 16 points in 3D world space, and threshold the predicted occupancies to determine voxel presence. Any grid cell whose occupancy probability is 0.5 or greater is considered occupied. We then apply an additional mask to the voxel grid, pruning any occupied voxels whose corresponding

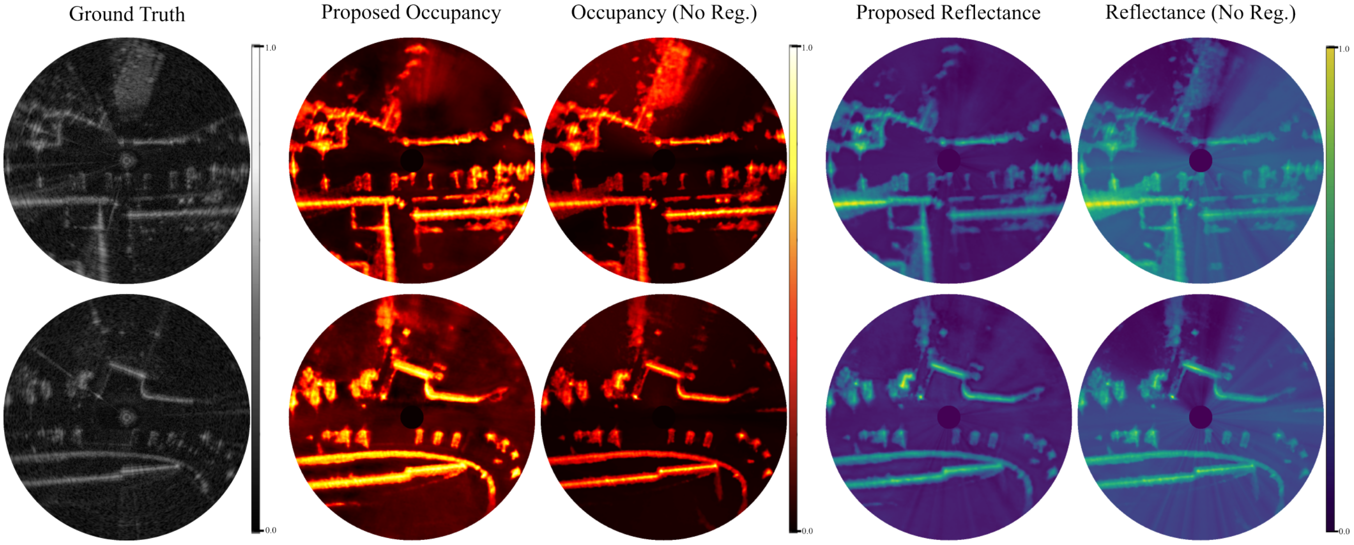

Figure 1: Ablation experiment demonstrating the benefits of regularizing our neural occupancy field with an estimated ground truth occupancy $O(P_r)$. The figure is divided into three sections, from left to right: the leftmost column displays the raw radar waveform; the middle section presents the predicted $\hat{\alpha_R}$ with (left) and without (right) $O(P_r)$ supervision; and the rightmost section illustrates the differences in predicted $\hat{\rho\gamma}$. Notably, the absence of additional occupancy supervision results in the absorption of specular highlights into the occupancy field and, therefore, the loss of its geometric interpretation. Supervising the occupied space directly helps to separate these specular effects into the reflectance field.

$\rho\gamma$ at that location is below a threshold. The motivation behind this last technique is that, at locations in space where the reflectance is too low, the sensor would be unable to discern any occupancy.

2 EVALUATION DETAILS

2.1 FFT Reconstruction

We evaluate signal reconstruction performance using PSNR and RSME on the FFT data. The FFTs are represented in Cartesian coordinates, centered at the radar location, and are three-channel tensors with normalized values in the 0 to 1 range. We compute PSNR and RSME between two images X and Y as

$$\mathrm{PSNR}(X,Y) = 10.0 \log\left(\frac{1}{\mathrm{MSE}(X,Y)}\right), \tag{9}$$

$$\mathrm{RSME}(X,Y) = \sqrt{\mathrm{MSE}(X,Y)}. \tag{10}$$

2.2 Ground-truth Point Cloud Generation

To evaluate the scene reconstructions of our method, we first generate a point cloud by thresholding the ground truth radar FFTs and accumulating the resulting points from all frames in the sequence. However, estimating ground truth geometry from raw waveform data is notably challenging due to low angular sampling, entangled multipath effects, and sensor noise. Consequently, we also derive ground truth scene geometry using LiDAR data, which provides a resolution an order of magnitude higher than that of radar. We aggregate LiDAR data across the entire sequence to establish a dense and accurate ground truth. This LiDAR-derived geometry is then refined by filtering and projecting it into the radar frame, where it is combined with the sparse, yet robust and informative radar point cloud. Specifically, we further refine this point cloud by retaining only those LiDAR points that are within a maximum distance of 0.1 m from a corresponding radar point. By doing so, we ensure that our evaluations of different methods are strictly confined to areas covered by both LiDAR and Radar sensors, allowing for a precise comparison of their effectiveness in scene reconstruction. This high resolution and refined point cloud is used only during evaluation and not for training the scene representation.

2.3 BEV Occupancy Reconstruction

The metrics used to evaluate the BEV Occupancy Reconstruction are the Chamfer Distance (CD) and Relative Chamfer Distance (RCD) between the generated 2D ground truth point cloud and the point cloud obtained by applying peak-finding to the predicted occupancies. Chamfer Distance between two point clouds X and Y is computed as the sum of the distances from each point in X to its nearest neighbor in Y, plus the sum of the distances from each point in Y to its nearest neighbor in X. We compute CD and RCD as

$$CD(X,Y) = \frac{1}{2}\left(\frac{1}{x}\sum_{x\in X} distPred_x + \frac{1}{y}\sum_{y\in y} distGT_y\right) \tag{11}$$

$$RCD(X,Y) = \frac{1}{2}\left(\frac{1}{x}\sum_{x\in X}\frac{distPred_x}{\|x\|_2^2} + \frac{1}{y}\sum_{y\in Y}\frac{distGT_y}{\|y\|_2^2}\right) \tag{12}$$

with $distPred_x = \min_{y\in Y}\|x-y\|_2^2$, $distGT_y = \min_{x\in X}\|x-y\|_2^2$.

The capture setups for our experiments exhibit variations in the Field of View (FoV) for both LiDAR and Radar sensors. Notably, the radar provides data across a complete 360° field, whereas the LiDAR focuses more densely on a narrower 120° field, primarily directed

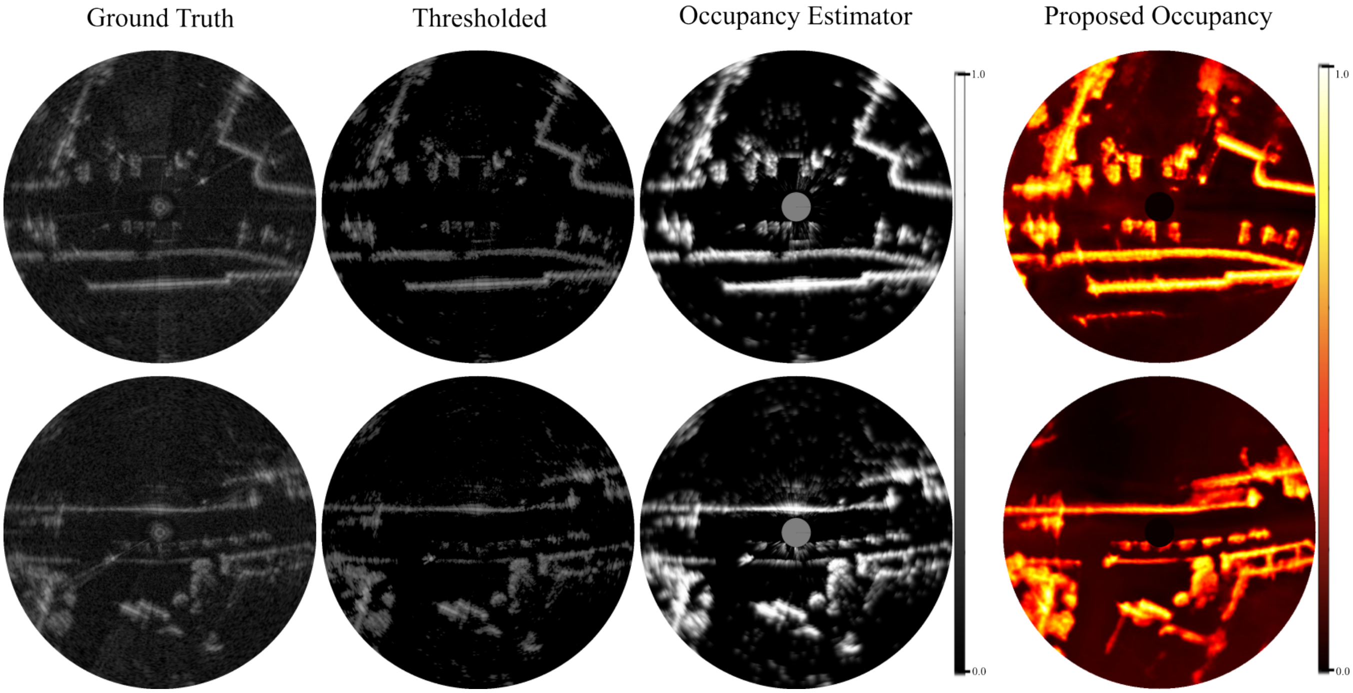

Figure 2: Visualization of $O(P_r)$. from left to right, the sequence begins with the raw input FFT P_r, followed by the thresholding results P_r', the estimated $O(P_r)$, and finally the our neural occupancy field prediction $\hat{\alpha_R}$. The thresholding effectively removes arbitrary noise and specular reflections from P_r. The final estimate $\hat{\alpha_R}$ not only correlates well with $O(P_r)$ but also achieves sharper detail, especially at greater distances, as it can integrate occupancy signals from across frames to recover a more complete scene representation. For example, note how most vehicles are only partially resolved in $O(P_r)$, while $\hat{\alpha_R}$ captures their silhouettes more clearly.

forward. This discrepancy in FoV coverage results in certain regions where the LiDAR and radar data do not overlap. In these non-overlapping areas, the refined ground truth point cloud may contain empty spaces due to the necessity of matching data from both sensor types. Therefore, when matching predicted radar points to the nearest accumulated refined ground truth point to compute distance metrics, we exclude any predicted points that have no ground truth points within 2 m. This way, we do not penalize radar points outside the field of view of the LiDAR. However, this is no longer an issue when matching ground truth points to predicted radar points. As such, we only apply the 2 m maximum in the former case, and not the latter case.

3 ADDITIONAL EXPERIMENTS

In this section, we present the results of additional experiments performed with our method.

3.1 Decomposed Scene Reconstructions

In our model, we formulate predicted radar FFT intensity via a physically-based decomposition of radar cross section σ into projected cross-sectional area α and one joint reflectivity and directivity term $\rho\gamma$. This introduces helpful inductive biases to improve model performance, and our model is able to disentangle the two directly from radar FFT data. See Fig. 3 for examples of this decomposition. Note that in these scenes, highly metallic objects such as cars and walls have high predicted reflectance.

Here we provide additional results that validate the proposed method for reconstructing 2D and 3D geometries within outdoor environments on multiple scenes in Fig. 4. Traditional radar point clouds, which have undergone post-processing, offer a sparsity that compromises the precision of scene reconstruction. Attempts to deduce bird's eye view occupancy from these point clouds using a grid mapping technique [Werber et al. 2015] do not successfully capture accurate geometry. In contrast, our method capitalizes on the unprocessed frequency-space radar data, facilitating the production of high-fidelity bird's eye view occupancy, and remarkably precise 3D geometry, despite the original data stemming from a standard 2D radar scan. The absence of physics-based ray importance sampling leads to a marked decrease in the quality of predicted occupancy, to the extent that smaller entities, such as vehicles, may not be consistently detectable. Moreover, without ray super-sampling, our model cannot learn a unified 3D scene representation. This result serves to corroborate the validity of the modeling approach we have introduced.

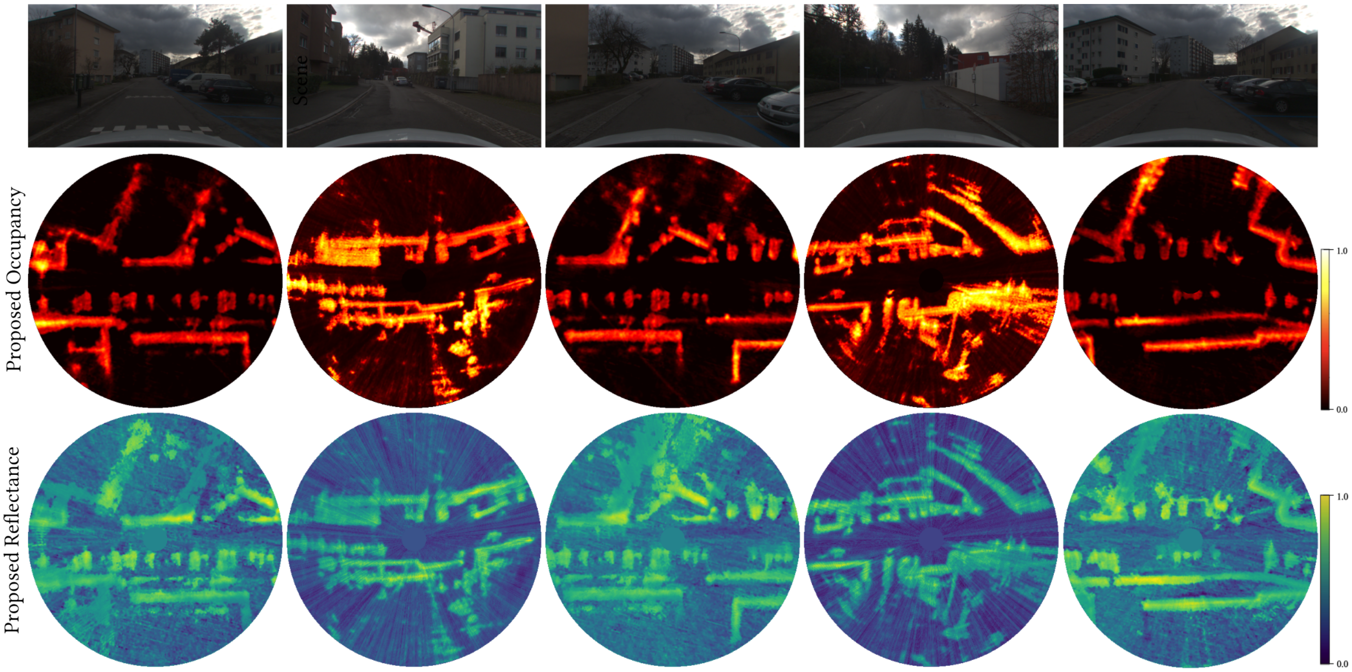

Figure 3: Decomposed occupancy and reflectance prediction. We report the predicted decomposition results on different scenes (columns). Our proposed physics-based method disentangles the radar cross section σ into a geometry-dependent occupancy term α (2nd row) and a reflectance representation term $\rho\gamma$ (3rd row). By separating geometry from view-dependent material reflectance, our model is capable of reconstructing the physical structure of a scene while also generating synthetic radar signatures, a task that would be unfeasible without this disentanglement.

4 ADDITIONAL ACQUISITION AND DATASET DETAILS

In this section, we provide additional information about the acquisition setup that we mount on our test vehicle, and the dataset we acquire with this setup.

Acquisition Setup. We list detailed sensor specifications in Table. 1. Our custom acquisition setup is housed in a 3D computer-aided design (CAD) representation of our waterproofed sensor setup in Fig. 5. Two metal plates are mounted directly to the roof of the car. The front plate provides support for the LiDAR and the waterproof box. The rear plate supports the radar and is elevated compared to the front one due to the natural curvature of the car roof. All sensors are affixed to these plates using adhesive screws. The waterproof box is constructed from metal with a plexiglass front. To align the camera sensors vertically, the RGB camera is elevated with an additional mounting piece. The processing unit and other electrical components are secured on stationary steel in the car trunk.

We utilize an NVIDIA Jetson Xavier AGX as the processing unit for our sensors, it offers an 8-core ARM CPU with 32GB RAM at a speed of 137GB/s. We store all raw data streams directly to an onboard 2TB SSD and do all post-processing at a later stage, separated from the recordings. The processor communicates with the sensors using ROS 1, employing ROS wrappers provided by the manufacturers to capture individual data streams.

Table 1: Multi-modal Sensor Setup. We capture a multi-modal dataset with a radar sensor capable of capturing RAW radar data, a LiDAR sensor, camera and IMU sensor. These are the specifications of the sensorset mounted on the test vehicle, as described in the main manuscript.

Modality	Name	Specifications
Radar	Navtech CIR-DEV	4 Hz, range: 330 m, range resolution: 43.8 mm, horizontal angular resolution: 0.9°
LiDAR	RS-LiDAR-M1	10 Hz, range: 200 m, avg. angular resolution: 0.2°, HFOV: 120°, VFOV: 25°, 75K points/scan
Camera	TRI023S-CC	30 Hz, 8-bit RGB, 1920×1080, HFOV: 77°, VFOV: 43°
GNSS&IMU	simpleRTK2B Fusion	30 Hz, KF fusion, RTK accuracy: <10cm,

Our recording platform operates on a separate 12V battery, independent of the car's electrical system. While the camera, the Jetson AGX, and the LiDAR operate on 12V, the radar requires 24V. To address this disparity, an extra power converter is incorporated to increase the battery output voltage specifically for the radar.

To minimize blurring of the lens caused by water droplets, a hydrophobic coating was applied to both the acrylic glass window of our waterproof box and the LiDAR cover glass, and they were regularly wiped clean when recording in adverse weather.

Synchronization. We synchronize all internal clocks of our different sensors according to the procedure detailed below. Our computer for recording data is connected to an online GPS time server

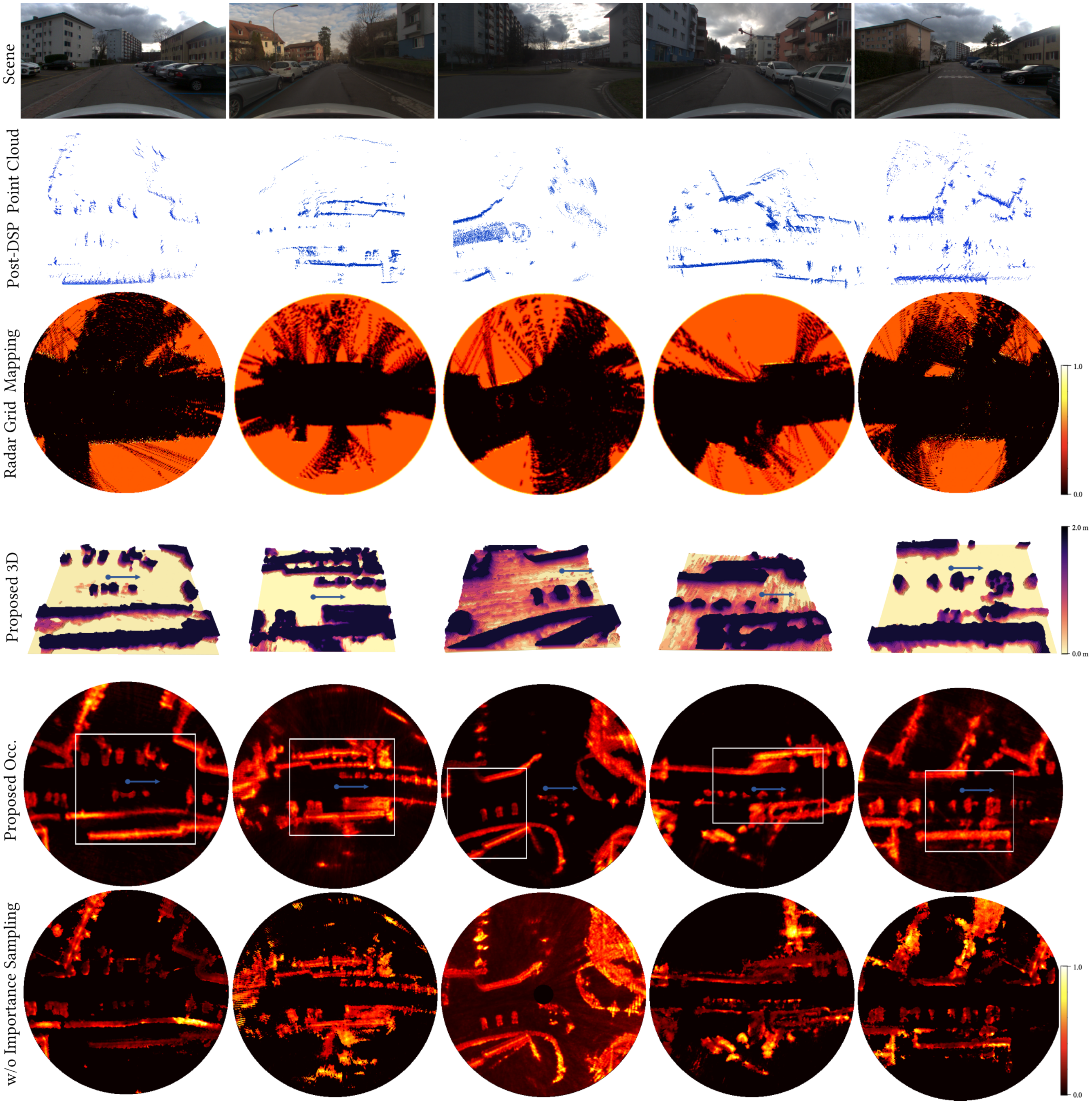

Figure 4: Additional Results of Radar Fields for Scene Reconstruction. We further assess the proposed method for the reconstruction of 2D and 3D scene geometry in outdoor scenes (columns). Conventional post-processed radar point clouds, accumulated in the second row, are too sparse to provide accurate scene reconstruction. Reconstructed bird's eye view occupancy produced by a grid mapping [Werber et al. 2015] from these radar point clouds fails to recover accurate geometry. The proposed method relies on raw frequency-space raw radar measurements and achieves high quality bird's eye view occupancy (fourth row), and even accurate 3D geometry (third row) although the measurement itself is a conventional 2D radar scan. Notice that the reconstructed voxel grid can also accurately represent the incline of the ground in the scenes in column 3 and column 4. Without physics-based ray importance sampling (last row), the predicted occupancy decreases substantially such that smaller objects like vehicles are not always resolved, validating our proposed model.

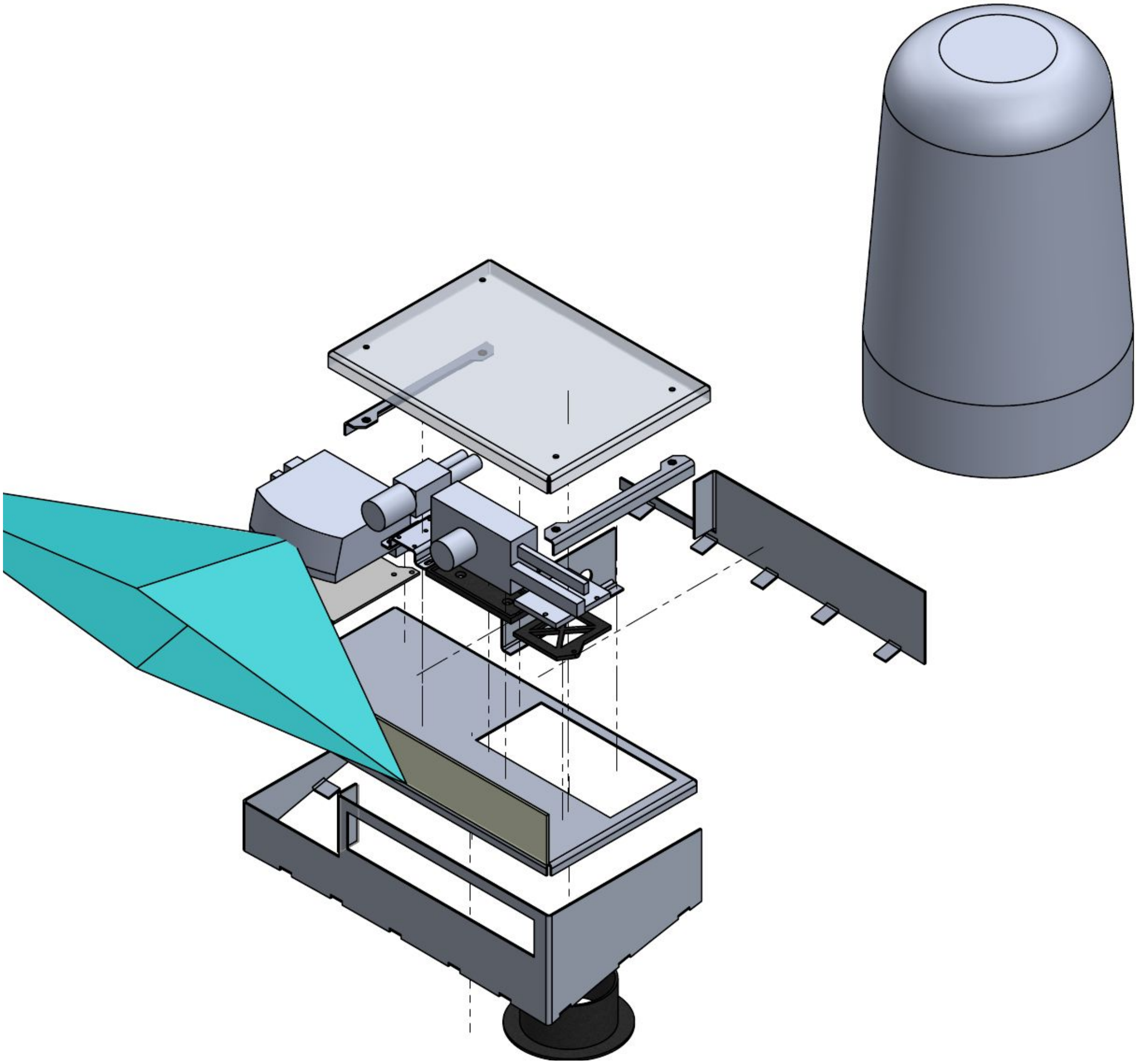

Figure 5: Waterproof Sensor Setup. The front-facing LiDAR on the left with its stylized viewing cone and the separate scanning radar on the right, as well as the GNSS antenna are waterproofed and mounted outside of the waterproof box. The camera and GNSS/IMU module are mounted in a metal box with a plexiglas window towards the front. The metal box has additional cable throughlets for the LiDAR, radar, and GNSS antenna cables, and is positioned lower than the radar to ensure that the radar scans remain undisturbed.

through the network time protocol (NTP) and serves as the primary clock. The radar is synchronized using NTP. The LiDAR and camera are aligned using the precision time protocol (PTP) along with software timestamping. The GNSS clock naturally synchronizes with atomic clocks on GPS satellites.

Sensor Modalities. We further discuss the various sensors included in our recording platform and their respective strengths and weaknesses:

Radar: Radar is a radiolocation system using radio waves. It is an active sensor and its large wavelength makes it mostly invariant to small particles encountered in most weather phenomena. It excels

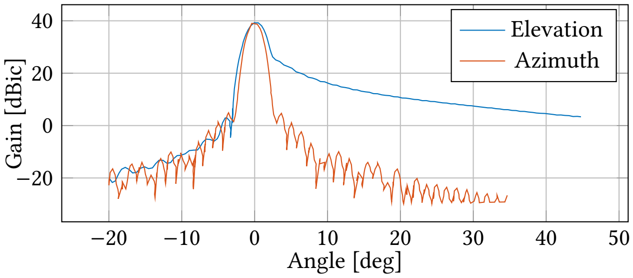

Figure 6: Angular-Dependent Antenna Gain Response. We rely on the antenna gain to recover 3D information via our physics-based ray importance sampling from raw 2D radar scans. We plot antenna gain in decibels for our Navtech CIR-DEV radar sensor. Specifically, gain is plotted as a function of azimuth and elevation offset, in degrees, relative to the center of any given transmitted radar beam.

in detecting objects over extensive distances and remains resilient in challenging weather conditions like rain, snow, and fog, making it the most reliable automotive sensor in adverse situations. This has led to a growing trend of incorporating radar sensors into driving datasets. However, it is essential to note that radar often suffers from lower resolution and various noise modes. Each recorded radar frame is stored as a 7300 × 400 PNG file. In a similar fashion to [Barnes et al. 2020], we store all radar beam metadata in the first 8 pixels of the images. The metadata includes the azimuth, a valid flag, and a timestamp split into seconds and nanoseconds. The radiation profile of our sensor is plotted in Fig. 6.

LiDAR: Light detection and ranging (LiDAR) is a commonly-used sensor for autonomous vehicles and functions by emitting a 905nm wavelength laser pulse that travels to an object and reflects back to the sensor, providing dependable distance and intensity measurements in both daytime and nighttime conditions. LiDAR is an active sensor that is invariant to illumination changes. However, because the laser pulse travels the distance to an object twice - once to the object and once back to the sensor - LiDAR measurements degrade more than camera measurements in situations with reduced visibility, like fog or rain. We store each LiDAR point individually with its timestamp. This allows for detailed data processing, like ego-motion compensation, but is very storage intensive.

Camera: The fundamental sensor in most perception systems is the RGB camera. It is cost-effective, widely adopted, and offers good spatial resolution with a high frame rate compared to other sensors. Despite these advantages, the camera is a passive sensor that requires well-illuminated scenes for optimal performance. Specifically, its performance is considerably restrained under challenging visual conditions such as night, rain, fog, and snowfall, resembling the limitations of human vision. Additionally, exposure to fast changes in light intensity may result in under- or oversaturation of the sensor. We store each camera frame as a 1920 × 1080 RGB image.

Additional Dataset Statistics. Our dataset offers a comprehensive exploration along three key dimensions, enriching our understanding of diverse external influences on model performance. Specifically, we recorded both day and night scenes, capturing the nuances of varying lighting conditions. In addition, we documented urban and parking lot scenarios, presenting distinct challenges in scenes and vegetation differences. Moreover, our dataset encompasses both clear weather and foggy scenes, providing insights into the impact of adverse weather conditions on perception systems. Whereby our final selection of 15 sequences includes 12 sequences from daytime, 3 from nighttime, 10 from urban settings, 5 from parking lots, 11 under clear weather, and 4 under foggy conditions. This intentional diversity allows for a nuanced examination of how different contextual factors influence model performance. Each of these in-the-wild sequences averages 17.5 seconds with 70 radar frames, 150 LiDAR frames, and 450 camera frames.

Dataset Preprocessing. We preprocess our data to compute trajectories and transformations between all of our sensor spaces and world coordinates. For the radar recordings, we apply a low threshold to the raw FFT intensity returns to effectively reduce background noise. To enhance the reliability of the GNSS data, a Kalman filter is introduced with a linear forward model. This filter acts to ensure smoother positions by effectively mitigating noise and minimizing smaller jumps present in the GNSS recordings.

Additional Dataset Samples. We provide additional recorded scenes in Fig. 7 - 8. Fig. 7 displays exemplary day scenes with three urban and two parking lot scenes with vegetation. Both radar and LiDAR capture the surrounding scenes with their multiple vehicles well. Fig. 8 displays exemplary night scenes with 3 parking lot scenes and two highly adverse foggy night scenes. We can already recognize less of the scenes with our camera, but the LiDAR and radar give similar results to our day scenes. The fog reduces the visual conditions significantly, resulting in a blurry camera image and a degraded point cloud that only reaches a few meters in range. The radar, on the other hand, is unaffected by these adverse conditions.

5 LIMITATIONS

This method relies on raw radar data, which is inherently a limitation, as it is not always made available by radar sensors and may require hardware modifications and custom readout routines to handle the large data volume. In the future, we hope raw radar data will be made available more broadly by sensor manufacturers. Currently, our method cannot be deployed in real-time. However, cross-modal initialization and recent acceleration methods for neural fields may further improve inference time.

REFERENCES

Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. 2020. The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset. In *IEEE International Conference on Robotics and Automation (ICRA).*

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–15.

Klaudius Werber, Matthias Rapp, Jens Klappstein, Markus Hahn, Jürgen Dickmann, Klaus Dietmayer, and Christian Waldschmidt. 2015. Automotive radar gridmap representations. In *2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM).* 1–4. https://doi.org/10.1109/ICMIM.2015.7117922

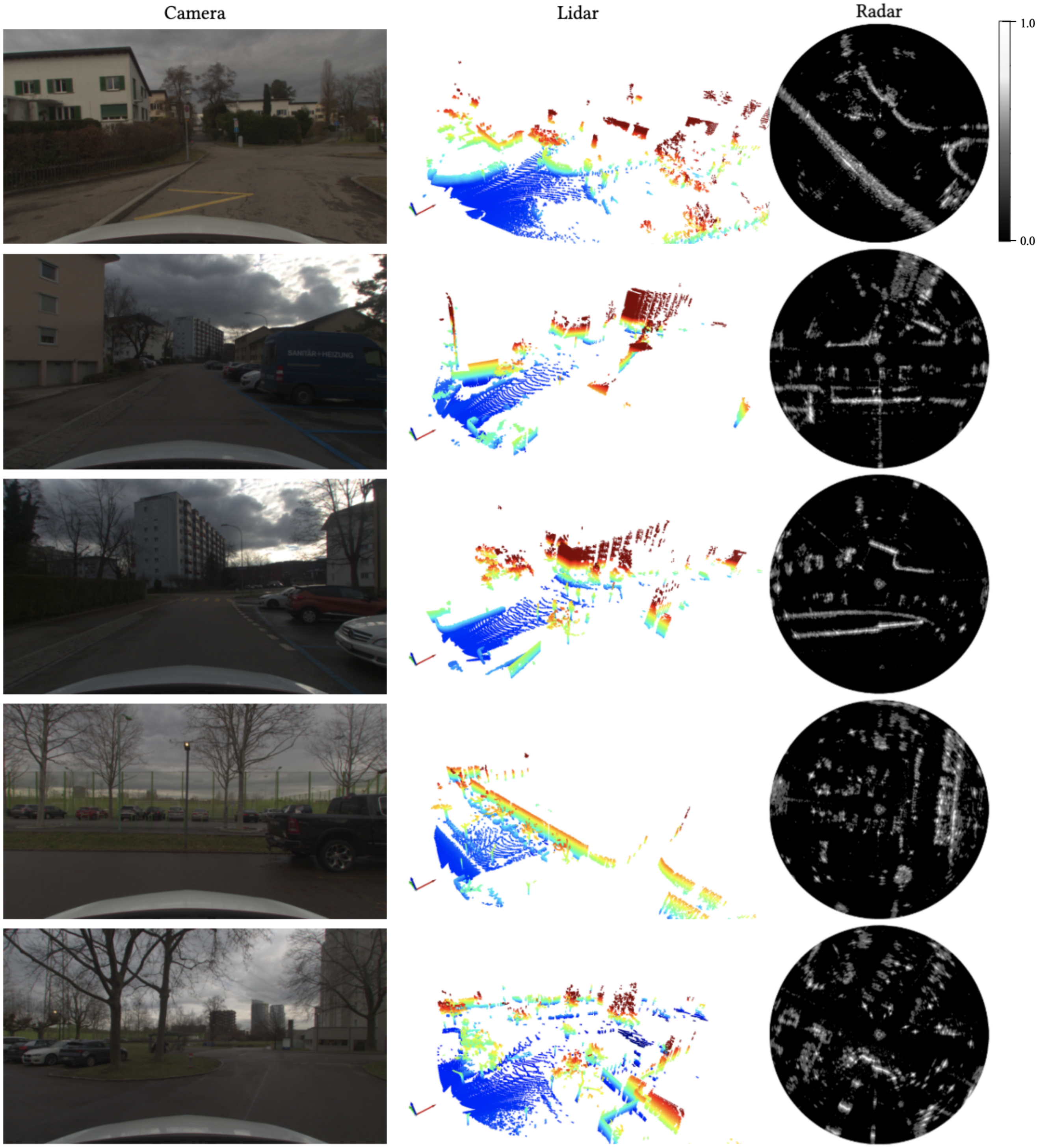

Figure 7: Additional exemplary scenes with (1st column) images from our forward-facing RGB camera, (2nd column) point clouds from our forward-facing LiDAR, color-coded by height, with the car in the bottom-left, and (3rd column) 40-meter-radius BEV radar returns, with the car in the center and pointing to the right.

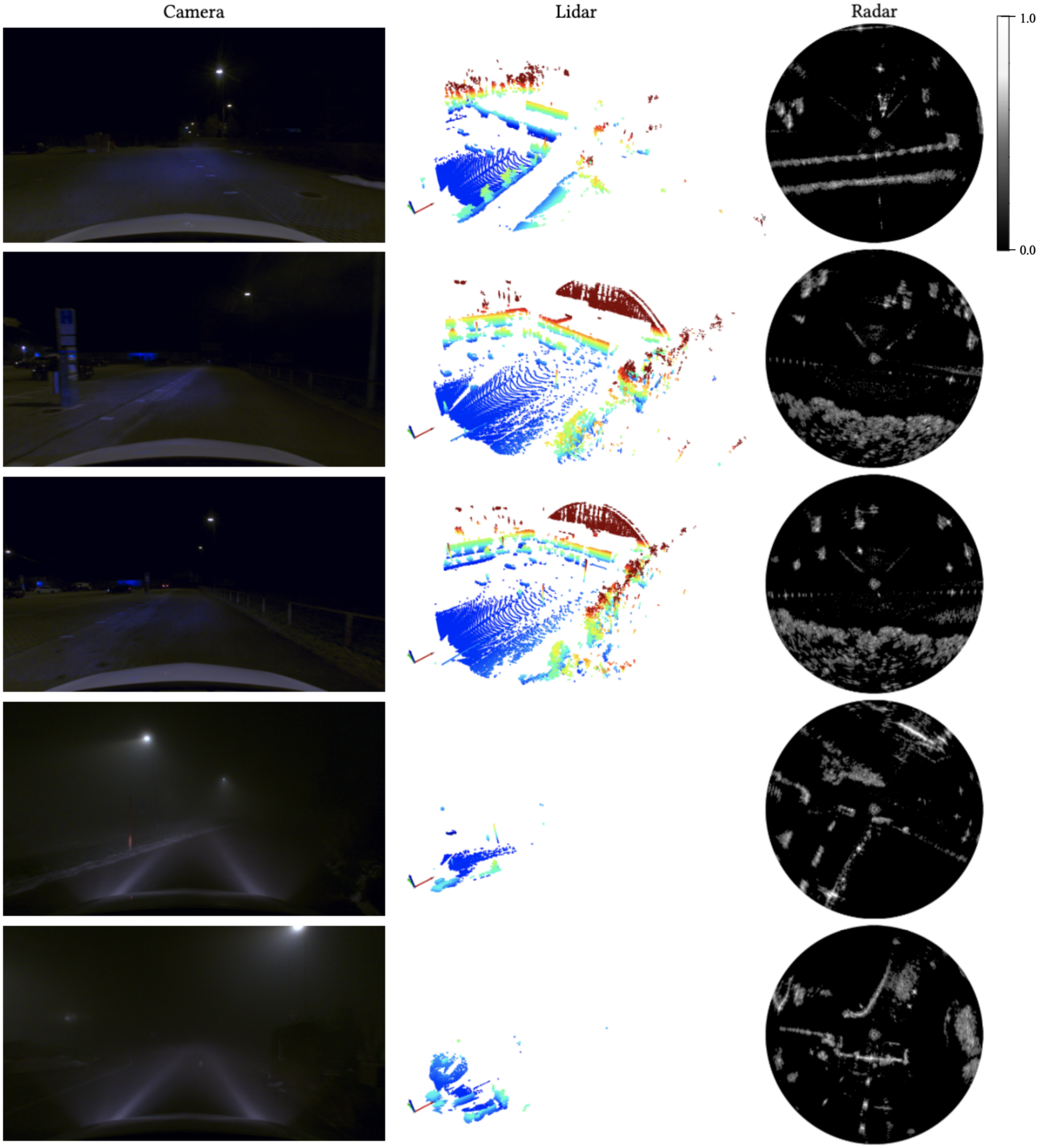

Figure 8: (Continued) Additional exemplary scenes with (1st column) images from our forward-facing RGB camera, (2nd column) point clouds from our forward-facing LiDAR, color-coded by height, with the car in the bottom-left, and (3rd column) 40-meter-radius BEV radar returns, with the car in the center and pointing to the right.