RadSimReal: Bridging the Gap Between Synthetic and Real Data in Radar Object Detection With Simulation

Oded Bialer* and Yuval Haitman*

General Motors, Technical Center Israel

oded.bialer8@gmail.com, haitman@post.bgu.ac.il

Abstract

Object detection in radar imagery with neural networks shows great potential for improving autonomous driving. However, obtaining annotated datasets from real radar images, crucial for training these networks, is challenging, especially in scenarios with long-range detection and adverse weather and lighting conditions where radar performance excels. To address this challenge, we present RadSimReal, an innovative physical radar simulation capable of generating synthetic radar images with accompanying annotations for various radar types and environmental conditions, all without the need for real data collection. Remarkably, our findings demonstrate that training object detection models on RadSimReal data and subsequently evaluating them on real-world data produce performance levels comparable to models trained and tested on real data from the same dataset, and even achieves better performance when testing across different real datasets. RadSimReal offers advantages over other physical radar simulations that it does not necessitate knowledge of the radar design details, which are often not disclosed by radar suppliers, and has faster run-time. This innovative tool has the potential to advance the development of computer vision algorithms for radar-based autonomous driving applications. Our GitHub: https://yuvalhg.github.io/RadSimReal.

1. Introduction

Automotive radar plays an important role in autonomous driving systems, offering long-range object detection capabilities and robustness against challenging weather and lighting conditions. The radar emits radio frequency (RF) signals and, through the processing of reflected echoes from the surrounding environment, creates a radar reflection intensity image [6]. The image contains reflection intensities

---
*Both authors contributed equally to this work.

Both authors are with General Motors, Yuval Haitman is also with the School of Electrical and Computer Engineering in Ben Gurion University of the Negev.
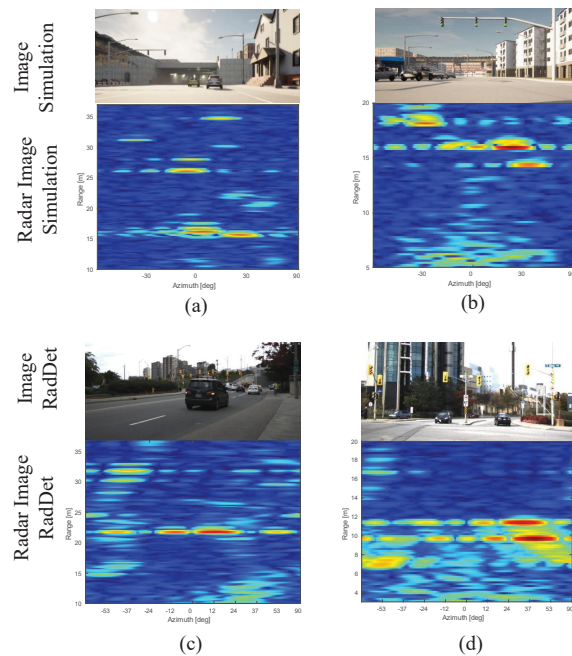
Figure 1. Comparison between synthetic and real radar images from four different scenarios. Each scenario shows the camera image and the corresponding radar image. (a) and (b) simulation scenarios. (c) and (d) real scenarios.

corresponding to range and angle coordinates, providing a visual representation of the scene. Afterwards, computer vision algorithms are employed to identify objects within this visual image.

Numerous Deep Neural Network (DNN) methods have emerged for detecting objects in radar images [11, 22, 25, 30, 44, 45]. These techniques involve training the DNN using annotated real data. Several datasets containing real annotated radar images have been introduced [27–29, 36, 40, 44]. These datasets vary in terms of the radar type and environmental conditions. However, the primary challenge with object detection DNNs trained on real data lies in the considerable effort required to collect and annotate the data. This challenge is particularly hard in the case of radar since it is used to detect objects at long range, adverse weather

and lightening conditions in which annotations are difficult to obtain.

In an effort to address the challenge posed by data annotations, an alternative approach that generates training data through generative methods has been proposed. Several studies have explored the training of a Generative Adversarial Network (GAN) using unlabeled real radar data to produce synthetic radar data that closely mimics actual real radar data [10, 16, 42, 43]. These studies have demonstrated that when training a detection DNN with synthetic data generated by the GAN and testing on real data, the performance gap compared to training with real data is small.

Despite the advantage of not requiring data annotation, generative data generation still presents the hurdle of collecting a large volume of unlabeled real data. This poses a significant limitation in system development, as it necessitates collecting a substantial domain-specific dataset for each unique radar sensor, distinct sensor mounting conditions, and environmental conditions in order to effectively train the GAN for generating radar images that match the specific distribution of the data. This problem is resolved when using physical radar simulation instead of generative radar simulation.

In physical simulation, synthetic radar images are created through the physical modeling of the environment and the radar sensor [3, 33]. Consequently, for each distinct radar sensor, mounting setup, scenario distribution, and environmental conditions a simulated dataset can be generated without the necessity of collecting any real radar data. Physical radar simulations have been extensively explored in the radar domain [3, 20, 21, 33, 35, 38]. Their process involves several steps. It begins with the creation of 3D automotive scenarios, followed by the calculation of radar reflections achieved through signal ray tracing from the radar to objects and back. Subsequently, the radar's received signal is generated based on these reflections and the specific radar hardware configuration. Finally, the radar image is produced by applying radar-specific signal processing algorithms to the received signal.

The specific hardware and signal processing design of a radar significantly influence the output radar image. Thus, the domain shift from one radar type to another is significant, possibly more pronounced than in other sensors like cameras. This underscores a major limitation of current physical radar simulations, as they demand a comprehensive understanding of the radar's hardware parameters and signal processing algorithms to produce a radar image that accurately replicates the real-world image. These details are not always disclosed by radar suppliers, and even when available, their implementation within the simulation necessitates radar expertise. Another issue with physical radar simulation is its high computational demand, resulting in long processing times when generating large datasets.

The similarity between real radar images and synthetic images generated by physical radar simulation, when the hardware and signal processing details of the radar are available, has been evaluated both qualitatively [20, 21, 35] and quantitatively. The quantitative evaluation involves calculating correlation coefficients between synthetic and real images [33], and measuring distances between corresponding reflection points or objects in synthetic and real images [26]. However, there has been a lack of evaluation concerning the performance gap between an object detection DNN trained with physical radar data and tested on real data, in comparison to a DNN solely trained and tested on real data.

In this paper, we present *RadSimReal* an innovative physical radar simulation method that does not require prior knowledge of the radar hardware specifications or its signal processing algorithms. Consequently, it can be applied to a wide range of radar sensors without necessitating expertise in radar-specific details. Additionally, this novel simulation approach offers considerably faster processing time in comparison to conventional physical radar simulations. Fig. 1 illustrates the similarity between radar images generated by *RadSimReal* and real radar images, which raises the question of whether synthetic images can replace real ones for training object detection neural networks. To answer this, we conduct a novel analysis comparing the performance of DNN object detection models trained with *RadSimReal* data and tested on real data versus models trained and tested exclusively with real data. Our findings reveal that object detection DNNs trained on *RadSimReal* data and tested on real data exhibit performance levels comparable to those trained on real data when both the training and testing datasets are sourced from the same real dataset, and outperform them when the training and testing are from different real datasets. *RadSimReal* is a powerful tool for efficiently generating extensive annotated training data with flexibility in radar sensor type, mounting configurations and environmental conditions, all without the overhead effort of collecting real data for each specific setting.

The main contributions of the paper are as follows:

1. A pioneer analysis revealing that object detection DNNs, trained with physical radar simulation data and tested on real data, perform comparably to models trained on real data when the train and test sets of the real data are from the same dataset, and outperforms them when they are from different datasets. This highlights the benefits of physical simulation, efficiently generating training data with annotations for diverse radar configurations without the challenges of collecting real data, overcoming a main limitation in generative approaches.

2. An innovative physical radar simulation technique that offers an important advantage over other reference physical simulation methods by not necessitating in-depth knowledge of specific radar implementation details,

which are often undisclosed, nor radar expertise, and also has a significantly faster run-time.

3. A simulation tool offering efficient data generation with ground truth information that conveniently supports various radar types. This tool will contribute to advancing radar-based computer vision research.

2. Related Work

2.1. Radar Object Detection

Various studies in the literature extensively investigate DNN methods for object detection in radar images. Kim et al. [22] applied YOLO [30] to radar images, surpassing the performance of conventional radar detection methods. Xu et al. [11] used a ResNet-18 encoder with a CNN decoder to estimate 3D bounding box properties. RADDet [44] integrated residual blocks and YOLO detection heads [7], while Zhang et al. [45] employed a CNN-based version of U-Net for radar object detection. Meyer et al. [25] utilized graph convolution networks for radar object detection. A two-stage object detection approach was introduced in [17]. An alternative approach for object detection involves using radar point clouds [9, 13, 15, 24, 32, 34], which are generated from the reflection points detected in radar images using the Constant False Alarm Rate (CFAR) algorithm [31]. CFAR introduces significant information loss during subsequent DNN processing [22], leading to a considerable degradation in object detection performance when compared to using radar images. Therefore, this paper focuses on DNN processing of radar images.

2.2. Radar Datasets

Numerous publicly accessible automotive radar datasets offer real radar images along with object annotations. The RADIATE [36] and Oxford RobotCar [4] datasets use a 360° mechanical scanning antenna, differing from conventional radar images generated by antennas arrays. The CRUW dataset [41] provides radar images with a limited range of up to 25 meters, and the annotations consist of object center points without bounding box information. RADIal [29] and K-Radar [28] provide high-resolution radar reflection intensity images, but details about their radar hardware are undisclosed, making them unsuitable for conventional physical radar simulation. The RADDet [44] dataset features radar images from 15 diverse automotive scenarios captured using a Texas Instruments (TI) automotive radar prototype [1, 2]. The CARRADA dataset [27] comprises radar images from 30 controlled scenarios, utilizing the same TI radar as RADDet, allowing for cross-dataset performance evaluation. Furthermore, the availability of hardware specifications for the TI prototype radar in these datasets facilitates the generation of synthetic data through traditional physical radar simulation.

2.3. Generative Radar Data Generation

Weston et al. [42] introduced a GAN method that generates synthetic radar images conditioned on a 3D representation from CARLA simulation [12], which was trained on a relatively large dataset comprising 222,420 images from the Oxford RobotCar [4]. Their study showed that a segmentation model, trained on synthetic data and tested on real data, achieves performance similar to a model trained and tested exclusively on real data. L2R GAN [39] used the same extensive dataset to train a GAN for converting LIDAR point clouds into radar images. Synthetic image quality was evaluated against real images using PSNR and SSIM metrics, along with qualitative assessment through a human subjective study. Oliveira and Bekooij [10] employed a GAN to generate synthetic radar images conditioned on bounding box layouts. They showed a small performance gap between an object detection DNN trained on synthetic data and tested on real data versus one trained solely on real data. Fidelis [16] used a GAN to generate radar received signals and derived radar images through signal processing. Their study showed a close distribution resemblance between synthetic and real images, assessed by the FID metric [19]. Wheeler et al. [43] used a Variational Autoencoder conditioned on an object list and raster grid of the ground roadways. They validated synthetic image similarity to real images using K-L divergence for clutter and average squared deviation for objects. While generative methods bridge the synthetic-real data gap, they require extensive training data for each radar type, mounting configuration, and environmental condition, posing a significant overhead.

2.4. Physical Radar Simulation Data Generation

In physical simulation, synthetic radar images are generated by simulating the environment, the radar sensor, and its installation on the vehicle. Numerous studies have proposed methods for physically modeling both the environment and radar systems. MaxRay [3] created realistic scenarios in Blender [18], simulated reflections through ray tracing with RF propagation properties, and tested radar detection and clutter removal on synthetic data of an OFDM radar simulation. ViRa [33] used the Unity game engine [8] to generate environments and simulated an FMCW radar. They demonstrated the similarity of their simulated radar data to real-world radar measurements in a laboratory setting using correlation metrics. Thieling et al. [38] proposed a radar simulation with environmental influences like rain. Their study assessed the simulation's performance in qualitative terms. In another approach [35], a ray tracing technique for a MIMO radar simulation was presented, with the realism of the generated radar images evaluated through qualitative comparisons to real radar images for the same scenarios. Several other ray tracing radar simulations for realistic automotive scenarios have also been published

[14, 20, 21], which verified the simulations' accuracy by comparing them to real measurements.

All these radar simulation studies haven't explored the use of DNN detection methods with physical radar simulation data. The performance gap between training object detection DNNs with radar simulation synthetic data versus real data and testing on real data remains unexplored. Furthermore, implementing these simulations requires detailed knowledge of radar hardware design and signal processing algorithms, factors often undisclosed by radar suppliers.

3. Proposed Physical Simulation Method

Fig. 2 presents a block diagram comparing our proposed physical simulation approach, *RadSimReal*, with the conventional physical simulation method. The diagram consists of three primary components. *RadSimReal* encompasses Fig. 2(a) and (c), whereas the conventional simulation method involves Fig. 2(a) and (b). Fig. 2(a) pertains to the environmental simulation, a shared element in both approaches, independent of the specific radar sensor. In this part, a 3D scene is generated using a graphics engine; in our implementation, we employed CARLA [12]. Then, dense reflection points from objects within the scene are acquired by ray-tracing of RF propagation paths, starting from the radar to objects in the environment and then returning to the radar [35]. Subsequently, the RF reflectivity of these points is determined using physical formulas [5], accounting for surface material, orientation, and distance from the radar. Following the environmental simulation, the subsequent step involves the radar simulation part, responsible for translating the intensities of reflection points into the radar's output image. Fig. 2(c) illustrates the radar simulation block diagram of *RadSimReal*, which deviates from the conventional radar simulation method shown in Fig. 2(b).

In the conventional radar simulation presented in Fig. 2(b), the radar's received signal, stemming from all reflection points within the scene, is acquired based on the specific hardware design details of the radar. Additionally, noise is introduced into the received signal. For detailed information regarding this part, we refer readers to the supplementary material. Following this, radar signal processing algorithms are applied to the received signal, yielding the radar 3D tensor. This tensor encapsulates the radar reflection intensity across range, azimuth angle, and Doppler cells. Notably, we consider a radar with only azimuth angle and without elevation, although the simulation is not limited to such radar and could be expanded to include elevation. In the final step of Fig. 2(b), the 3D tensor is transformed into an image format with dimensions of range and azimuth angle by selecting the maximum value along the Doppler dimension. It's worth mentioning that the simulation provides flexibility in outputting the entire radar tensor or converting it into an image using various techniques, depending on the

implementation of the object detection DNN.

The conventional radar simulation method depicted in Fig. 2(b), necessitates an in-depth comprehension of the radar hardware design and the associated software processing to faithfully mimic a specific radar. This entails a comprehensive understanding of specific details such as the transmit signal waveform, antenna array configuration, sampling rate, beamforming algorithms, and more. However, this presents a major challenge, as these details are often proprietary and not publicly disclosed by radar manufacturers. This limitation is resolved in Fig. 2(c) by *RadSimReal*, as elaborated in the following.

RadSimReal relies on the radar's point spread function (PSF). Every reflection point is apparent in the radar tensor by a multi-dimensional PSF, spanning dimensions of range, angle, and Doppler, and centered on the coordinates of the reflection point. Fig. 3(a) shows an example of a 2D slice of a radar's PSF in range and angle centered at a range of $25m$ and an azimuth angle of $0°$. The PSF exhibits a wide spread in the angular domain but a narrow spread in both range and Doppler, although the Doppler aspect is not depicted in the figure. This characteristic arises from the radar's coarse angular resolution and high range and Doppler resolution. It is essential to note that the shape of the PSF depends upon the specific radar hardware design and signal processing algorithms [37]. Fortunately, it can be acquired from a radar tensor of a narrow object, such as a pole or a dedicated corner reflector [23]. Importantly, obtaining the PSF does not require knowledge of any details about the radar design, thereby circumventing a key limitation of conventional radar simulation. The straightforward procedure for obtaining the radar's PSF from a radar tensor is detailed in the supplementary material.

In Fig. 3(b), we present a radar image generated through the conventional simulation outlined in Fig. 2(b). This image illustrates a scenario with three reflection points and is free from noise. In Fig. 3(c), we depict the outcome of the conventional simulation for the same scenario, this time with the addition of noise. The white points in the figures denote the locations of these reflection points. Notably, the image displays a superposition of the radar's PSF centered on each point. Therefore, the same image can be obtained by convolution between the reflection points and the PSF. As a result, *RadSimReal* generates the radar tensor by 3D convolution of the reflection points with the PSF, followed by the addition of noise, as presented in Fig. 2(c). The tensor is then converted to an image similarly as in Fig. 2(b).

Despite the differences in the simulation implementations in Fig. 2(c) and Fig. 2(b), they lead to the generation of an identical radar image when the full PSF is applied. For a detailed mathematical explanation of this equivalence, please refer to the supplementary material. Notably, the energy of the PSF diminishes rapidly from its central point in
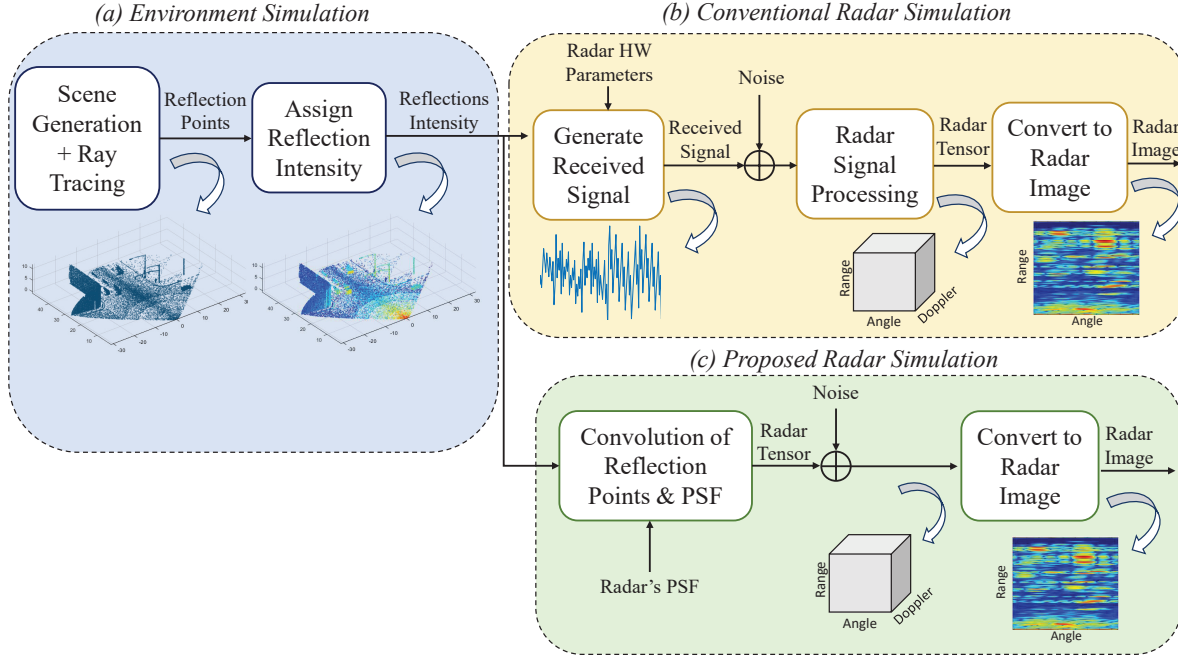
Figure 2. Block diagram illustrating the processing steps for conventional simulation (a)+(b) and *RadSimReal* (a)+(c). (a) Simulates the environment to generate reflection points with RF reflectivity of an automotive scene, while (b) and (c) represent the conventional approach and *RadSimReal*'s approach, respectively, for transforming the reflection points into a radar image.

both the range and Doppler domains. To reduce the computational complexity of the simulation, we truncate the PSF to encompass up to 99% of its energy. This truncation leads to a substantial reduction in the PSF size by a factor exceeding 1000, thereby significantly enhancing the run-time of the convolution operation in the simulation. In Fig. 3(d), the truncated PSF is displayed, containing 99% of the energy of the original PSF from Fig. 3(a). The radar image, obtained through convolution of the truncated PSF with the same three reflection points as depicted in Fig. 3(b), is showcased in Fig. 3(e) without noise and in Fig. 3(f) with the addition of noise. The truncated portion of the PSF has very low intensity value (about 80 dB lower than PSF peak), which is significantly lower than the noise level. Consequently, the difference between the radar image from the conventional simulation (Fig. 3(c)) and our proposed simulation utilizing the truncated PSF (Fig. 3(f)) is practically indistinguishable.

While *RadSimReal* (Fig. 2(a)+(c)) produces radar images similar to the conventional simulation (Fig. 2(a)+(b)), it possesses two significant advantages over the conventional simulation. Firstly, it eliminates the necessity of possessing in-depth radar design information by relying on a simple radar measurement of its PSF and noise variance, as detailed in the supplementary material. Secondly, it exhibits a significantly faster run-time, as evaluated and demonstrated in Section 3.2.

3.1. Simulation Fidelity Evaluation

In this section, we assess the resemblance between real radar images and synthetic images generated by *RadSim-Real*. Our evaluation initiates with a comparison between synthetic and real radar images generated from the same scenario. Creating a simulation scene that faithfully replicates a real-world scenario for which an actual radar measurement was taken poses a considerable challenge. To overcome this challenge, we substituted the reflection points derived from the CARLA simulation engine in Fig. 2(a) with points obtained from a high-resolution LIDAR sensor. This LIDAR was positioned in close proximity to the radar and captured measurements from the same scene at the same time as the real radar measurement. For the assignment of the RF reflection intensity to the LIDAR points the surface material and orientation of the LIDAR points were obtained by the following three steps: (1) Manual segmentation of LIDAR points into object types such as vehicles, poles, signs, roads, and buildings. (2) Surface material allocation of each point based on its object type. (3) Computation of the angle of the surface normal vector using a polygon mesh generated from the LIDAR points. The remaining stages of the simulation processing were as outlined in Fig. 2(c).

An example of a real radar image is shown in Fig. 4(c) and a synthetic radar image in Fig. 4(d) generated from the same scene (as explained above). The prototype radar used
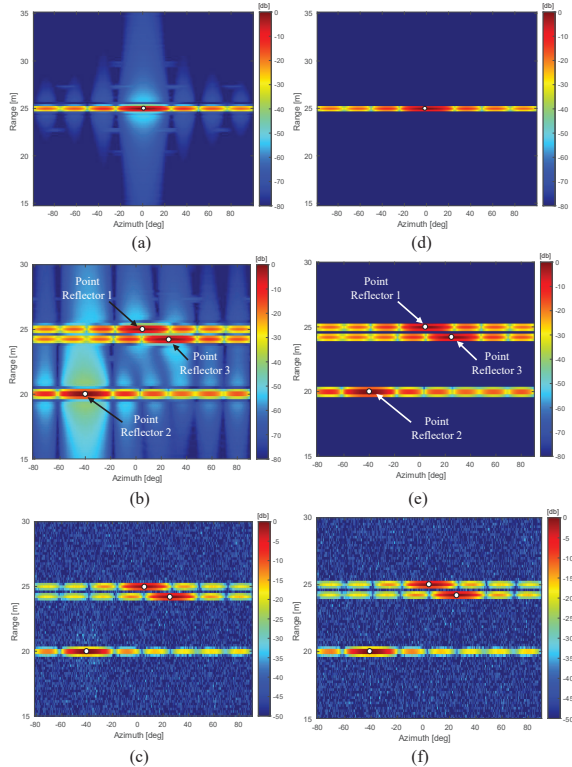
(a)

(d)

(b)

(e)

(c)

(f)

Figure 3. Radar image generated with conventional simulation vs. *RadSimReal*. (a) Radar's PSF 2D slice in range and angle dimensions. (b) Radar image without noise generated by conventional simulation for a scenario with 3 reflection points. (c) Radar image of (b) with noise. (d) Truncated PSF with 99% of its energy. (e) Radar image obtained for the same scenario as in (b) by *RadSimReal* without noise, (f) The radar image of (e) with noise.

in Fig. 4(c) had $3.9°$ azimuth resolution, and $0.28m$ range resolution, and the same radar was simulated in Fig. 4(d). The camera image of the scene is presented in Fig. 4(a), and the LIDAR points segmented to different object types are shown in Fig. 4(b). It is observed that the synthetic and real reflection intensity images have close resembles, which shows that the simulation models well the real radar.

Next, we present a comparison between real radar images extracted from the RADDet dataset [44], and synthetic images generated by simulating the same radar as in RADDet with *RadSimReal*. As the RADDet dataset lacks LIDAR measurements, simulating the precise scenarios of real radar images, as depicted in Fig. 4, is unfeasible. Hence, we present synthetic and real radar images from different scenarios and assess their characteristic resemblance. In Fig. 1, the upper two rows illustrate synthetic radar images along with their corresponding camera images, produced by simulating the radar used in the RAD-Det dataset. Meanwhile, the lower two rows exhibit real radar images and their respective camera images from the

(a) Camera image

(b) Segmented LIDAR points

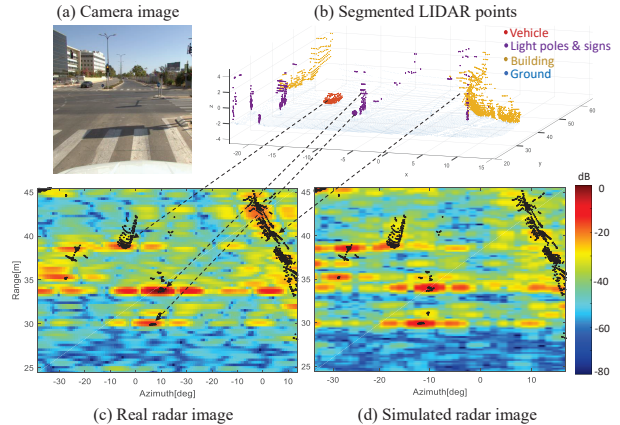(c) Real radar image

(d) Simulated radar image

Figure 4. Comparison between *RadSimReal* image and a real radar images for the same scenario. (a) Camera image of the scenario. (b) High-resolution LIDAR points segmented by object type. (c) Real radar image. (d) *RadSimReal* image. The black points in (c) and (d) represent the LIDAR points.

RADDet dataset. Although the synthetic and real images stem from different scenarios and cannot be compared one to one, they exhibit analogous characteristics. Both real and synthetic radar images display reflections with varying intensities and have a similar spreading functions. The close resemblance between the synthetic and real images makes it hard to distinguish between them, which is another indication that *RadSimReal* models well the real radar.

We also assessed the statistical similarity between the *RadSimReal* data and real data using the Frechet Inception Distance (FID) [19]. The FID score is commonly employed in the literature to quantify the statistical resemblance between synthetic and real datasets. The FID score between the RADDet training set (comprising 8196 images) and the RADDet test set (comprising 1962 images) is 6.76. On the other hand, the FID score between the RADDet training set and a synthetic dataset of 10,000 images generated by *RadSimReal* is 6.54. The similarity between these two scores indicates that the statistical characteristics of the synthetic dataset produced by *RadSimReal* closely resemble with those of the RADDet data.

3.2. Computation Efficiency

The complexity ratio between conventional simulation and *RadSimReal* corresponds to the ratio between the entire radar tensor volume and the PSF volume. As explained in Section 3, our simulation truncates the PSF to preserve 99% of its energy, drastically reducing its volume. This results in a substantial complexity reduction, approximately by a factor of 1000, with *RadSimReal* compared to conventional radar simulation. Further details on the computational complexity calculation and run time measurements are available in the supplementary material.

4. Simulation to Real Domain Gap Analysis

In this section, we analyze the object detection performance gap between models trained with *RadSimReal* data and those trained with real data, both tested on real data. For this analysis we use three different object detection methods: 'U-Net', 'RADDet', and 'Probabilistic'. 'U-Net' employs a U-Net as proposed in [45] with an additional input channel of the input image Cartesian coordinates. 'RADDet' refers to the object detection network introduced in the RADDet paper [44], while 'Probabilistic' is the network from [11].

We assess performance using the RADDet [44], CAR-RADA [27], and CRUW [41] datasets. These datasets feature automotive radar reflection images captured with a TI radar prototype [1, 2] in diverse scenes, providing extensive testing coverage. While RADDet and CARRADA feature 2D bounding box annotations for objects, CRUW provides annotations indicating the center points of objects, to which we've subsequently added bounding box extensions using the CFAR algorithm [31]. RADDet includes images from 15 densely populated automotive scenarios with favorable weather conditions, while CARRADA comprises 30 staged scenarios with varying object densities and weather conditions, including challenging conditions like snow. In both datasets the radar is mounted on a stationary platform. In the CRUW dataset the radar is mounted on a vehicle, capturing radar images from scenarios where the ego-vehicle was in motion (highway and city streets) and scenarios where it was stationary (campus roads and parking lots).

We employed a dataset split for RADDet that ensures different scenarios in the train and test sets thereby preventing potential overfitting in the split proposed in the dataset paper [44]. Our RADDet split consists of a training set with 8196 images and a test set with 1962 images. For CAR-RADA, the training set comprises 2208 images, and the test set includes 276 images. As for CRUW, our training set comprises 9623 images, with a test set of 2226 images. The synthetic dataset generated by *RadSimReal* comprised 10000 training images, comparable in size to that of RAD-Det and CRUW. The simulated scenarios involved a radar mounted on a vehicle driving in city streets, experiencing both stationary and moving phases.

We conducted performance tests on the three object detection models mentioned above using three separate test datasets: RADDet, CARRADA and CRUW. These models were individually trained with four distinct datasets: the RADDet training set, the CARRADA training set, the CRUW training set, or the *RadSimReal* dataset. Performance was assessed using Average Precision (AP) for class 'car' at IOU thresholds 0.1, 0.3 and 0.5. The AP was determined by the area under the precision-recall curve. Table 1 presents AP results, comparing performance between training and testing on individual datasets versus training with *RadSimReal* and testing on different datasets. It addi-

tionally includes cross-dataset evaluation between RADDet and CARRADA, both characterized by a stationary radar setup (unlike CRUW), with the primary difference lying in their scenes. The results reveal several important insights. All three models trained using *RadSimReal* exhibit performance on both the RADDet and CRUW test sets that closely resembles their performance when trained on the corresponding RADDet or CRUW training sets. In evaluations with the CARRADA test set, models trained with *RadSimReal* consistently outperform those trained with the CARRADA training set, likely due to the small size of the latter. Notably, models trained with RADDet experience a significant performance decline on the CARRADA test set compared to their performance when trained with *RadSimReal*. These results demonstrate that object detection DNNs trained with *RadSimReal* perform comparable to those trained on real data and even outperform DNNs trained on real data when subjected to cross-dataset evaluation or when dealing with limited training data.

Subsequently, we assess the performance of object detection models on the RADDet test set when trained using a combination of data from *RadSimReal* and the training set of RADDet. The outcomes of this evaluation are presented in Table 2. The findings reveal that augmenting the *RadSimReal* dataset with real datasets from RADDet does not yield a significant performance enhancement. This suggests that the domain shift from *RadSimReal* data to real data is insignificant.

Next, we provide qualitative examples comparing between an object detection model trained with *RadSimReal* and one trained with real data. In Fig. 5, we showcase the detection scores at the output of the 'U-Net' model for two examples taken from the RADDet test set, comparing the performance of the model trained with *RadSimReal* against the same model trained with the RADDet training set. Each example is displayed in a separate row, featuring the original image alongside the output detection scores and bounding boxes of both models. Notably, the detection scores and boxes of both models resemble each other, indicating that the model trained with synthetic data delivers similar performance to the one trained exclusively on real data.

The analysis in this section shows *RadSimReal*'s success in bridging the object detection performance gap between synthetic and real data. It is important to note that the images generated by *RadSimReal* are similar to those produced by other existing physical radar simulations. Consequently, training object detection models with other physical radar simulations could achieve a similar performance as with *RadSimReal*. The significance of our study lies in unveiling this key discovery for the first time and introducing *RadSimReal*, which holds advantages over existing simulations. It eliminates the need for in-depth knowledge of the radar design, typically undisclosed, and has faster run-

Table 1. AP at different IOU for three object detection models trained on *RadSimReal* or real data and tested on real data

| Test Set	Train Set	U-Net Model			Probabilistic Model			RADDet Model		
		@0.1	@0.3	@0.5	@0.1	@0.3	@0.5	@0.1	@0.3	@0.5
RADDet	RADDet	84.76	83.01	55.53	83.31	74.80	40.68	83.69	72.96	47.95
	RadSimReal	85.63	82.16	57.64	82.75	75.83	52.36	83.48	73.91	46.63
	CARRADA	22.33	19.46	14.04	24.58	22.53	16.98	19.18	15.37	9.47
CARRADA	CARRADA	50.99	49.00	44.65	50.16	48.22	39.51	31.94	26.65	17.79
	RadSimReal	70.77	62.47	43.96	63.51	56.92	41.07	72.39	65.65	28.76
	RADDet	62.40	56.84	30.78	57.02	54.23	19.53	69.63	61.63	20.49
CRUW	CRUW	86.66	78.83	56.54	81.90	77.22	52.95	85.74	70.10	54.36
	RadSimReal	86.82	77.51	55.47	80.04	76.50	51.45	86.21	69.60	52.48
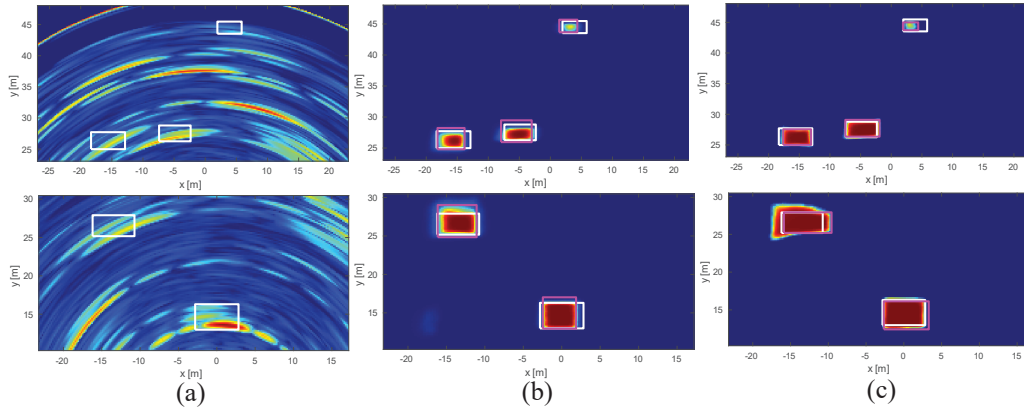
Figure 5. Qualitative comparison of object detection DNN trained on *RadSimReal* vs. real data from the RADDet dataset. Rows correspond to different scenarios from RADDet test set. (a) Input radar image, (b) 'U-Net' model's detection score and bounding boxes trained with RADDet. (c) 'U-Net' trained with *RadSimReal* data. Detected and ground truth bounding boxes marked in pink and white, respectively.

Table 2. Object detection AP at various IOU for 'U-Net' trained with combined *RadSimReal*'s data (Sim) and real data (Real)

Test Set	Train Set	@0.1	@0.3	@0.5
RADDet	Sim	85.63	82.16	57.64
	Sim + Real	86.09	83.38	58.13
CARRADA	Sim	70.77	62.47	43.96
	Sim + Real	71.11	62.63	44.52

time. Additionally, we acknowledge that the synthetic to real performance gap can also be closed by generating data with generative methods such as GAN. However, these approaches have a major drawback compared to the physical simulation model; they require the collection of a large amount of real data for each distinct variation in radar type, its mounting configuration and environmental conditions.

5. Conclusion

This paper introduces *RadSimReal*, a novel physical radar simulation that generates synthetic radar images for train-

ing object detection DNNs. We have shown that the *RadSimReal* images closely resemble real radar images both qualitatively and statistically. Most importantly, our results reveal that training object detection DNNs with these synthetic images and testing them on real data yield results similar to those obtained when training exclusively with real data. Moreover, it attains superior performance in cross-dataset evaluations with different real datasets.

RadSimReal offers distinct advantages over alternative methods of synthetic data generation. It can efficiently simulate diverse radar types without the need for extensive real data collection, a process demanding substantial resources, or in-depth knowledge of proprietary radar implementation details, which are often confidential. Instead, it only requires a measurement of the radar's PSF. This work highlights the great potential of radar simulation in radar-based computer vision applications, paving the way for its increased adoption and further exploration in this field.

References

[1] AWR1843 data sheet, product information and support, https://www.ti.com/product/awr1843, . 3, 7

[2] TI mmwave-sdk software development kit (SDK), https://www.ti.com/tool/mmwave-sdk, . 3, 7

[3] Maximilian Arnold, M Bauhofer, Silvio Mandelli, Marcus Henninger, Frank Schaich, Thorsten Wild, and Stephan ten Brink. Maxray: A raytracing-based integrated sensing and communication framework. In *2022 2nd IEEE International Symposium on Joint Communications & Sensing (JC&S)*, pages 1–7. IEEE, 2022. 2, 3

[4] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438. IEEE, 2020. 3

[5] R Mahafza Bassem and Z Elsherbeni Atef. Matlab simulations for radar systems design. *CRC, London*, 2004. 4

[6] Igal Bilik, Oded Bialer, Shahar Villeval, Hasan Sharifi, Keerti Kona, Marcus Pan, Dave Persechini, Marcel Musni, and Kevin Geary. Automotive mimo radar for urban environments. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–6. IEEE, 2016. 1

[7] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3

[8] Ismail Buyuksalih, Serdar Bayburt, Gurcan Buyuksalih, AP Baskaraca, Hairi Karim, and Alias Abdul Rahman. 3d modelling and visualization based on the unity game engine–advantages and challenges. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4: 161–166, 2017. 3

[9] Andreas Danzer, Thomas Griebel, Martin Bach, and Klaus Dietmayer. 2d car detection in radar data with pointnets. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 61–66. IEEE, 2019. 3

[10] Marcio L Lima de Oliveira and Marco JG Bekooij. Generating synthetic short-range fmcw range-doppler maps using generative adversarial networks and deep convolutional autoencoders. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–6. IEEE, 2020. 2, 3

[11] Xu Dong, Pengluo Wang, Pengyue Zhang, and Langechuan Liu. Probabilistic oriented object detection in automotive radar. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 102–103, 2020. 1, 3, 7

[12] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 3, 4

[13] Maria Dreher, Emeç Erçelik, Timo Bänziger, and Alois Knol. Radar-based 2d car detection using deep neural networks. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020. 3

[14] Manuel Dudek, René Wahl, Dietmar Kissinger, Robert Weigel, and Georg Fischer. Millimeter wave fmcw radar system simulations including a 3d ray tracing channel simulator. In *2010 Asia-Pacific Microwave Conference*, pages 1665–1668. IEEE, 2010. 4

[15] Zhaofei Feng, Shuo Zhang, Martin Kunert, and Werner Wiesbeck. Point cloud segmentation with a high-resolution automotive radar. In *AmE 2019-Automotive meets Electronics; 10th GMM-Symposium*, pages 1–5. VDE, 2019. 3

[16] Eduardo C Fidelis, Fabio Reway, Herick Ribeiro, Pietro L Campos, Werner Huber, Christian Icking, Lester A Faria, and Torsten Schön. Generation of realistic synthetic raw radar data for automated driving applications using generative adversarial networks. *arXiv preprint arXiv:2308.02632*, 2023. 2, 3

[17] Yuval Haitman and Oded Bialer. Boostrad: Enhancing object detection by boosting radar reflections. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1638–1647, 2024. 3

[18] Roland Hess. *Blender foundations: The essential guide to learning blender 2.5*. Taylor & Francis, 2013. 3

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 3, 6

[20] 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. IEEE, 2017. 2, 4

[21] Martin Holder, Clemens Linnhoff, Philipp Rosenberger, and Hermann Winner. The fourier tracing approach for modeling automotive radar sensors. In *2019 20th International Radar Symposium (IRS)*, pages 1–8. IEEE, 2019. 2, 4

[22] Woosuk Kim, Hyunwoong Cho, Jongseok Kim, Byungkwan Kim, and Seongwook Lee. Yolo-based simultaneous target detection and classification in automotive fmcw radar systems. *Sensors*, 20(10):2897, 2020. 1, 3

[23] Eugene F Knott, John F Schaeffer, and Michael T Tulley. *Radar cross section*. SciTech Publishing, 2004. 4

[24] Florian Kraus, Nicolas Scheiner, Werner Ritter, and Klaus Dietmayer. Using machine learning to detect ghost images in automotive radar. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2020. 3

[25] Michael Meyer, Georg Kuschk, and Sven Tomforde. Graph convolutional networks for 3d object detection on radar data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3060–3069, 2021. 1, 3

[26] Anthony Ngo, Max Paul Bauer, and Michael Resch. A multilayered 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. IEEE, 2021. 2

[27] Arthur Ouaknine, Alasdair Newson, Julien Rebut, Florence Tupin, and Patrick Pérez. Carrada dataset: Camera and automotive radar with range-angle-doppler annotations. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5068–5075. IEEE, 2021. 1, 3, 7

[28] Dong-Hee Paek, Seung-Hyun Kong, and Kevin Tirta Wijaya. K-radar: 4d radar object detection dataset and benchmark for autonomous driving in various weather conditions. *arXiv preprint arXiv:2206.08171*, 2022. 3

[29] Julien Rebut, Arthur Ouaknine, Waqas Malik, and Patrick Pérez. Raw high-definition radar for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17021–17030, 2022. 1, 3

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 3

[31] Hermann Rohling. Radar cfar thresholding in clutter and multiple target situations. *IEEE transactions on aerospace and electronic systems*, (4):608–621, 1983. 3, 7

[32] Nicolas Scheiner, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. Radar-based road user classification and novelty detection with recurrent neural network ensembles. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 722–729. IEEE, 2019. 3

[33] Christian Schöffmann, Barnaba Ubezio, Christoph Böhm, Stephan Mühlbacher-Karrer, and Hubert Zangl. Virtual radar: Real-time millimeter-wave radar sensor simulation for perception-driven robotics. *IEEE Robotics and Automation Letters*, 6(3):4704–4711, 2021. 2, 3

[34] Ole Schumann, Markus Hahn, Jürgen Dickmann, and Christian Wöhler. Semantic segmentation on radar point clouds. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 2179–2186. IEEE, 2018. 3

[35] Christian Schüßler, Marcel Hoffmann, Johanna Bräunig, Ingrid Ullmann, Randolf Ebelt, and Martin Vossiek. A realistic radar ray tracing simulator for large mimo-arrays in automotive environments. *IEEE Journal of Microwaves*, 1(4):962–974, 2021. 2, 3, 4

[36] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. Radiate: A radar dataset for automotive perception in bad weather. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2021. 1, 3

[37] Merrill Ivan Skolnik. Introduction to radar systems. *New York*, 1980. 4

[38] Jörn Thieling, Susanne Frese, and Jürgen Roßmann. Scalable and physical radar sensor simulation for interacting digital twins. *IEEE Sensors Journal*, 21(3):3184–3192, 2020. 2, 3

[39] Leichen Wang, Bastian Goldluecke, and Carsten Anklam. L2r gan: Lidar-to-radar translation. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 3

[40] Yizhou Wang, Zhongyu Jiang, Yudong Li, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. Rodnet: A real-time radar object detection network cross-supervised by camera-radar fused object 3d localization. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):954–967, 2021. 1

[41] Yizhou Wang, Zhongyu Jiang, Yudong Li, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. Rodnet: A real-time radar object detection network cross-supervised by camera-radar fused object 3d localization. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):954–967, 2021. 3, 7

[42] Rob Weston, Oiwi Parker Jones, and Ingmar Posner. There and back again: Learning to simulate radar data for real-world applications. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12809–12816. IEEE, 2021. 2, 3

[43] Tim A Wheeler, Martin Holder, Hermann Winner, and Mykel J Kochenderfer. Deep stochastic radar models. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 47–53. IEEE, 2017. 2, 3

[44] Ao Zhang, Farzan Erlik Nowruzi, and Robert Laganiere. Raddet: Range-azimuth-doppler based radar object detection for dynamic road users. In *2021 18th Conference on Robots and Vision (CRV)*, pages 95–102. IEEE, 2021. 1, 3, 6, 7

[45] Guoqiang Zhang, Haopeng Li, and Fabian Wenger. Object detection and 3d estimation via an fmcw radar using a fully convolutional network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4487–4491. IEEE, 2020. 1, 3, 7