

Controllable Radar Simulation with Waveform Parameter Embedding

Weiqing Xiao^{1,7*} Hao Huang^{2,7*} Chonghao Zhong^{3,7*} Yujie Lin^{2,7} Nan Wang⁷ Xiaoxue Chen^{4,7}
Zhaoxi Chen⁵ Saining Zhang⁵ Shuocheng Yang⁶ Pierre Merriaux⁸ Lei Lei⁸ Hao Zhao^{4,7†}
¹NJU ²BJTU ³BIT ⁴AIR, THU ⁵NTU ⁶SVM, THU ⁷Lightwheel AI ⁸LeddarTech

weiqing001@smail.nju.edu.cn zhaohao@air.tsinghua.edu.cn

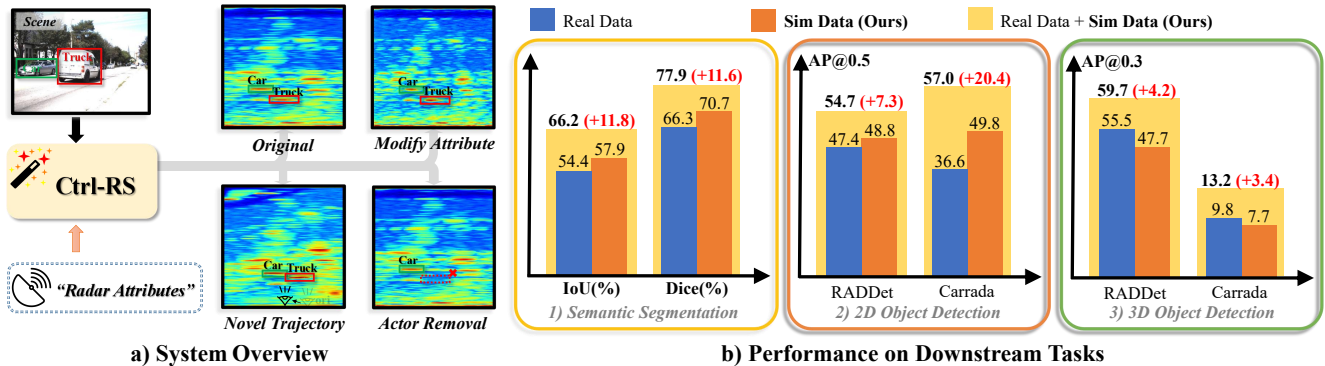


Figure 1. (a) Ctrl-RS enables controllable and realistic radar simulation by conditioning on customizable radar attributes. It supports flexible scene editing such as attribute modification, actor removal, and novel trajectories. (b) Ctrl-RS improves performance on various tasks including semantic segmentation, 2D/3D object detection. In all settings, Ctrl-RS’s synthetic data either matches or surpasses real data, and provides consistent gains when combined with real-world datasets.

Abstract

Autonomous driving simulators still lack high-fidelity radar, even though radar is critical for robust perception in adverse weather. A key obstacle is that raw radar point clouds are extremely sparse and stochastic, making it difficult to model; we argue that simulating the full range–azimuth–Doppler cube is a more principled target. Existing radar cube simulators either rely purely on neural generators, which are opaque and offer little control over sensor attributes, or on detailed electromagnetic pipelines, which are slow, require proprietary hardware specifications, and still struggle to capture real-world complexity. We introduce **Ctrl-RS**, a controllable radar cube simulation framework that combines the strengths of both worlds. First, we build an environment reflection tensor from diverse sensor sources (including LiDAR, monocular cameras, and existing radar). Second, we abstract radar physics into a compact set of waveform parameters that characterize the 3D point spread function, yielding an intuitive embedding of radar attributes such as range resolution, Doppler broadening, and azimuth beam shape. Third, we train a WARP-Net on a large mixed dataset that fuses real, analytically synthesized, and simulator-generated radar cubes to cover a wide distribution of radar attributes. Ctrl-RS

supports viewpoint changes, actor removal, and attribute editing. Experiments on RADDet, Carrada, and nuScenes show that our simulated data can match or surpass real radar in 2D detection and semantic segmentation, and consistently boosts performance in 3D detection when combined with real data. The Project is available at <https://github.com/zhuxing0/Ctrl-RS>.

1. Introduction

With its exceptional anti-interference capabilities and reliability under adverse weather conditions, radar plays a crucial role in advanced driver assistance systems (ADAS) [2, 8, 29, 41, 51, 55, 56]. However, its research and development have long faced challenges in data acquisition [26, 46, 62, 63]. Consequently, the importance of radar simulation has become increasingly prominent. Radar simulation not only replicates diverse environmental conditions and traffic scenarios but also enables researchers to explore different radar parameter settings to cope with the diversity and complexity of real-world applications.

Radar emits electromagnetic waves and processes the reflected echo signals from targets, generating a radar cube (typically in the form of a range-azimuth-Doppler tensor) that records the relative velocity (Doppler shift) and reflectivity at each position [13, 30, 45]. This provides a detailed

visualization of the surrounding environment [27, 49, 54, 57]. Existing radar simulation methods [1, 6, 16, 19, 42] can be broadly categorized into two types: generative radar simulation and physics-based radar simulation.

Generative radar simulation directly generates domain-specific radar cube slices [7, 18, 22, 53] through generative adversarial networks (GAN) [10, 42, 48] or variational autoencoders (VAE) [15, 52, 53]. Experimental results demonstrate that the generated synthetic data closely approximates real data in simple 2D tasks. However, this data-driven approach faces two major challenges: (1) Significant domain gaps exist in data distribution across different radar systems, necessitating separate data collection for each radar. (2) The absence of a physical prior prevents the extrapolation of radar attributes, limiting data simulation to predefined radar settings.

In contrast, physics-based radar simulation [6, 9, 24, 36, 58] simulates radar cube by modeling the complete process of an electromagnetic wave from emission to reception, ensuring interpretability and controllability over radar attributes. However, this approach demands precise radar hardware specifications, complex signal processing algorithms, and substantial computational time. Fortunately, recent progress [3] provides us with new insights, which characterize radar attributes by the standard 3D reflection signal (i.e., the reflection signal of a single reflection point, also called the point spread function or PSF), although it still needs to be measured individually for each radar.

In this paper, we present Ctrl-RS, a radar simulation method that can handle different radar attributes simultaneously (Fig 1), focusing on simulating the range-azimuth-Doppler tensor (i.e., the radar cube). We provide a complete radar simulation pipeline, carefully designed from environment simulation to radar cube simulation. Specifically, we explore and validate the feasibility of constructing radar reflection information from existing sensor data for environment simulation. Concurrently, we propose a Waveform-Aware Radar Prediction (WARP-Net) to achieve controllable radar cube simulation. In WARP-Net, the radar attributes are characterized by the waveform parameters of the standard 3D reflection signal, which is not only visually intuitive, but also makes it effective and easy for WARP-Net to capture the signal variations under different radar attributes. We propose a series of methods to construct a mixed dataset enriched with radar attributes to robustly train WARP-Net.

We not only demonstrate the accuracy of Ctrl-RS simulation results on different datasets, but also experiment with them in several downstream tasks (including semantic segmentation and 2D/3D detection). Extensive experimental results show that baseline models trained on our simulation data can easily achieve comparable or even better performance than models trained on real data. When combin-

ing Ctrl-RS’s simulation data with real-world datasets, all downstream tasks experimented in this paper achieve further performance improvements. Furthermore, our Ctrl-RS successfully achieves realistic and stable simulations under various sensor input conditions, as well as in novel viewpoints and edited scenarios, demonstrating its potential for generating new scenarios in autonomous driving applications.

2. Related Works

Physical radar simulation primarily obtains reliable simulation data by modeling each stage of electromagnetic wave propagation—from emitting and reflecting to receiving and processing [13, 30, 45]. For example, to simulate reflected echo signals, researchers have proposed a series of time-domain electromagnetic simulations [6, 9, 24, 36, 58], which model the behavior of electromagnetic fields over time by directly solving Maxwell’s equations. While these methods are physically accurate, they are constrained by high computational requirements and stability issues, making them impractical for real-time autonomous driving simulations. Alternatively, ray-tracing based radar simulations [1, 16, 19, 20, 44, 59] approximate signal propagation by modeling the interaction of electromagnetic waves with the environment. However, these methods require highly detailed hardware parameters and signal processing algorithms for the radar, which increases the complexity and computational requirements. Recently, RadSimReal [3] proposes to use standard 3D reflection signal (i.e., PSF) to construct a direct link from the environment to the radar cube, thus overcoming the reliance on radar-specific details and specific algorithms (see the Appendix § A.1 for a more detailed description). However, it still requires collecting data for each unique radar sensor and manually measuring the corresponding standard 3D reflectance signal. Our Ctrl-RS draws on this idea and proposes to use waveform parameters of standard 3D reflection signal in different dimensions to characterize the radar attributes, which in turn enables effective control of radar attributes.

Generative radar simulation aims to eliminate the need for detailed physical modeling of radar phenomena through a data-driven approach [11, 25, 38], thereby bridging the domain gap with real-world data. For example, a series of works [10, 48, 52] employ GAN or VAE to generate realistic radar cube slices. Additionally, several studies [4, 21, 28] have adopted NeRF to model 3D environments and generate novel view for radar. Despite these advances, radar signals are still significantly affected by physical factors such as radar sensor attributes and environmental conditions. Ignoring these factors may lead to non-corresponding results [20, 44, 59], i.e., the generated radar data does not satisfy a specific distribution in the real data. In contrast, our Ctrl-RS is a hybrid framework that not only achieves

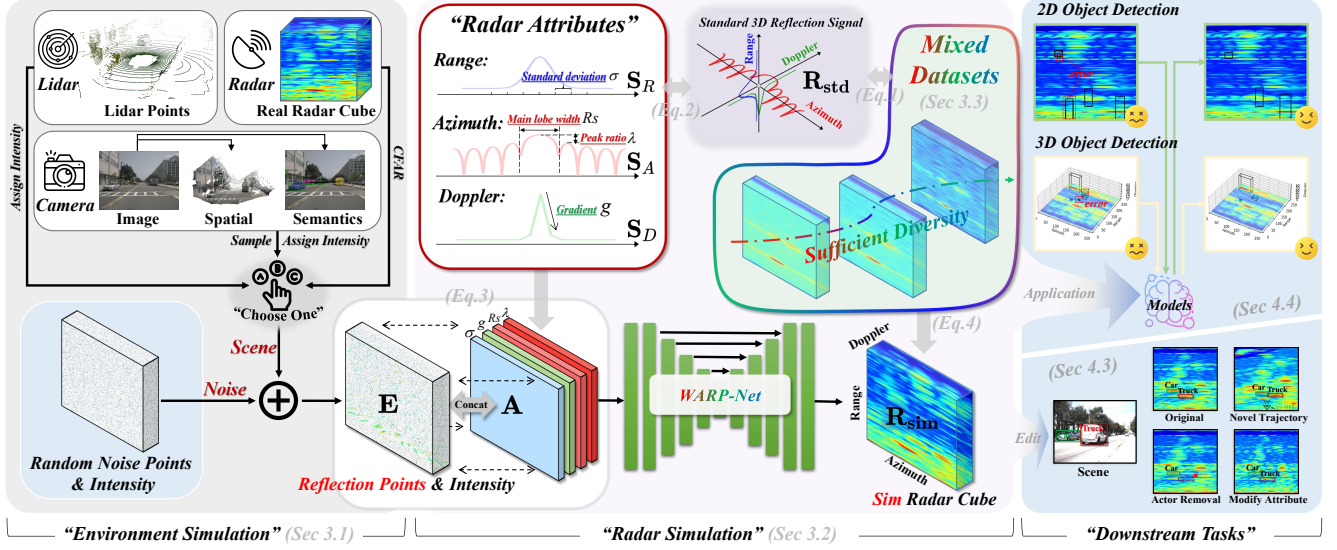


Figure 2. The overall framework of the proposed Ctrl-RS. The block diagrams from left to right show the environment simulation process, the radar simulation method and the downstream tasks of the simulation system in turn.

accurate generation of the complete radar cube through a data-driven approach (i.e., WARP-Net in this paper), but also incorporates physical prior (i.e., waveform parameters of standard 3D reflection signal) to capture radar signal variations under different radar attributes.

3. Method

As shown in Fig 2, our Ctrl-RS comprehensively designs the entire process from environment simulation (also known as environment construction) to radar simulation. In Sec 3.1, we first introduce environment simulation, a step traditionally regarded as a foundational module (independent of specific radar sensors) in prior methods. However, we explore a more flexible implementation approach that utilizes data from other sensors to achieve environment simulation. In Sec 3.2, we present the core module of Ctrl-RS, a controllable radar simulation method (WARP-Net) capable of directly simulating the radar cube (i.e., range-azimuth-Doppler tensor) of the current environment based on configurable radar attributes. In Sec 3.3, we describe how to construct an attribute-rich dataset for training WARP-Net.

3.1. Environment Simulation

The environment simulation aims to obtain all the reflection points (i.e., the points that reflect the echo signals) and reflection intensities in the scene. As shown in the left block diagram of Fig 2, in the environment simulation of Ctrl-RS, we construct a uniform reflection environment tensor $\mathbf{E} \in \mathbb{R}^{r \times d \times a}$, where r , d , and a denote the resolution of the radar cube in the Range-Doppler-Azimuth dimensions. In our approach, this tensor not only records the spatial locations, relative radar velocities, and reflection intensities of all reflective points in the current scene, but also considers

noise signals within the radar cube (essentially false echoes such as thermal noise or multipath scattering). Therefore, we divide the environment simulation into two parts: scene perception and noise simulation, and the specific details are described below.

Scene Perception. Unlike the traditional methods [3, 23, 31, 32, 34, 44] that strictly obtains reflection points through path tracing in graphics engines, we explore the feasibility of capturing scene reflection information through other sensors, such as LiDAR and camera. For example, we simply treat the LiDAR point cloud as reflected points and determine the reflection intensity of each point based on physical formulas [37]. Similarly, for camera image data, we use the reconstruction model VGGT [47] to obtain a dense scene point cloud. We then extract the target point cloud from it using instance masks generated by Grounded-SAM [43]. Finally, we obtain the reflected points and their reflection intensities through sparse random sampling and physical formulas [37]. Additionally, we support extracting reflection points and corresponding reflection intensities from existing radar data (using peak detection algorithms such as CFAR [14]). While reflection points obtained through these methods may lack precision, we experimentally verify Ctrl-RS’s robustness to them. This also implies the potential for directly supplementing radar information to a wide range of autonomous driving datasets.

Noise Simulation. Our key observation is that the noise reflection signal in the real radar cube is very similar to the scene reflection signal in terms of waveforms and is randomly distributed in the radar cube, as shown in Fig 3. Therefore, in order to more accurately reproduce the noise reflection signal in the radar cube, we propose to model the noise source using reflection points that are randomly

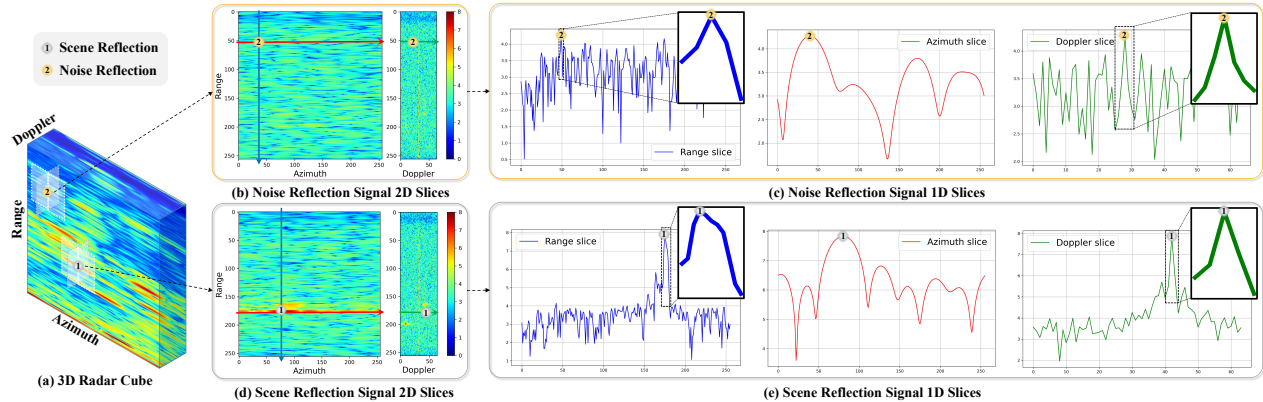


Figure 3. Visualization of 2D and 1D slices of radar cube from RADDet [60]. In the 1D slices of the Range and Doppler dimensions, we box the range of signals reflected from the reflection points, outside of the box the reflected signals come from the other noise reflection points not labeled in the figure.

distributed in all three dimensions: range, azimuth, and doppler (i.e., the noise signal is modeled as a radar signal from the noise reflection point, see the bottom left corner of Fig 2), instead of directly superimposing image noise (such as Gaussian noise) in the simulated radar cube as in the existing methods [3, 16, 19].

After obtaining the reflection points and their reflection intensities from the scene and the noise, we map each reflection point to a corresponding position in the reflection environment tensor \mathbf{E} , and set the value of that position to the specific reflection intensity of that point.

3.2. Controllable Radar Simulation

The middle block diagram in Fig 2 illustrates the radar simulation process of Ctrl-RS. The objective of this process is to simulate the radar cube based on the reflection environment tensor \mathbf{E} described in Sec 3.1 and to enable control over radar attributes. First, to overcome the need for radar-specific details in the simulation, we focus directly on the effect of different radars on the reflected waveform at each reflection point, and a waveform parameter-based representation of radar attributes (Sec 3.2.1). Then, combining the advantages of both physical radar simulation methods and generative radar simulation methods, we propose a hybrid approach, WARP-Net (Sec 3.2.2). On the one hand, it achieves efficient and realistic simulations through data-driven and implicit learning approaches. On the other hand, it effectively captures the variation of radar cubes under different radar attributes by integrating the reflection environment tensor \mathbf{E} and waveform parameter-based radar attribute embeddings.

3.2.1. Waveform Parameter-based Radar Attribute

Following RadSimReal [3], we model the radar cube $\mathbf{R} \in \mathbb{R}^{r \times d \times a}$ as a weighted superposition of standard 3D reflection signal \mathbf{R}_{std} (i.e., the PSF in RadSimReal [3]) at each

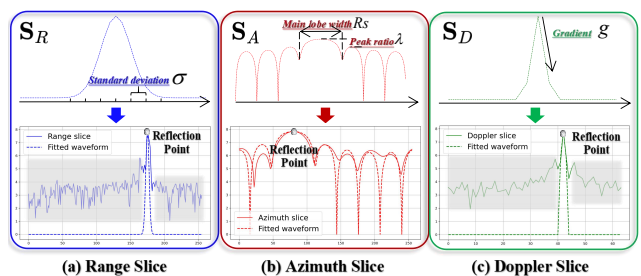


Figure 4. Visualization of the fitting functions and fitting results. Note that for the 1D slices, we only show the fitting results for a single reflection point, and the parts of the Range slice and Doppler slice that are covered with a transparent gray block actually belong to the other reflection points.

reflection point according to the reflection intensity:

$$\mathbf{R} = \sum_{i=1}^K I_i \cdot \mathbf{R}_{\text{std}}(r_i, d_i, a_i) \quad (1)$$

where I_i denotes the reflection intensity of the i -th reflection point, (r_i, d_i, a_i) denote the corresponding coordinates of the i -th reflection point in the radar cube, and K denotes the total number of reflection points. This modeling allows us to characterize radar attributes by the 3D waveform of \mathbf{R}_{std} without the need to obtain specific details of radar. Surprisingly, we observe that the 3D waveforms of \mathbf{R}_{std} are similar among different radar datasets [50, 60, 61] (see Fig A1 in Appendix). Inspired by this, we propose a waveform parameter-based radar attribute representation.

Specifically, we first fit slices of this 3D waveform in different dimensions using different functions (see Appendix § A.2 for function formulas):

$$\mathbf{R}_{\text{std}}(r_i, d_i, a_i) = \mathbf{S}_R(r_i) * \mathbf{S}_D(d_i) * \mathbf{S}_A(a_i) \quad (2)$$

where \mathbf{S}_R is the Gaussian function, capturing the Range resolution and decay. \mathbf{S}_D is the segmented linear function, modeling Doppler broadening. \mathbf{S}_A is the spectral function

of the window function, describing the Azimuth beam pattern. We then characterize the radar attributes by the standard deviation σ of \mathbf{S}_R , the gradient g of \mathbf{S}_D , and the main lobe width Rs and peak ratio λ in \mathbf{S}_A (theoretically, more parameters could characterize radar attributes more accurately, but in practice, these parameters are sufficient for simulation and are easily obtainable through measurement methods [33]). As shown in Fig. 4, by adjusting the parameters (σ , g , Rs , and λ), the proposed functions can accurately fit the 3D reflection signal \mathbf{R}_{std} at a single reflection point.

3.2.2. Controlling the Radar via Attribute Embedding

Unlike RadSimReal [3] which performs separate reflection waveform measurements for each type of radar, our goal is to train a radar simulation network with definable radar attributes. Thanks to our proposed waveform parameter-based representation of radar attributes (Sec 3.2.1), we can define radar attributes with a few simple parameters. On this basis, we propose WARP-Net, which directly simulates the radar cube based on the reflection environment tensor \mathbf{E} (Sec 3.1) and the waveform parameter-based radar attribute. Due to page constraints, we provide its architecture details in the Appendix § A.3. In summary, we first construct the radar attribute embedding $\mathbf{A} \in \mathbb{R}^{4 \times r \times d \times a}$, which is filled with σ , g , Rs , λ in order on its 4 channels. Then, we concatenate the radar attribute embedding \mathbf{A} and the reflection environment tensor \mathbf{E} , and process them with the WARP-Net:

$$\mathbf{R}_{\text{sim}} = U(\text{Concat}\{\mathbf{E}, \mathbf{A}\}) \quad (3)$$

where Concat denotes the Concat operation, and U denotes the WARP-Net (the final output is processed using the Relu function), and $\mathbf{R}_{\text{sim}} \in \mathbb{R}^{r \times d \times a}$ is the simulation radar cube.

3.3. Mixed Datasets and Model Training

In this section, we describe how to construct the training dataset (Sec 3.3.1) and the training strategies (Sec 3.3.2) for our WARP-Net. Full implementation details are provided in the Appendix.

3.3.1. Mixed Datasets

For the proposed WARP-Net, both the diversity of radar attributes and the quality of radar cube in the training dataset are crucial. However, the existing real radar datasets only cover a limited number of radar attributes. To overcome this challenge, we design two efficient data augmentation methods, creating a mixed dataset rich in radar attributes based on existing real-world radar data. Depending on the collection route, the mixed radar dataset can be divided into 3 parts: the real dataset A, the synthetic dataset B, and the simulation dataset C. The motivation and ideas for constructing these three parts are described below (see the Appendix § A.6 for specific details).

Real dataset A. It consists of multiple radar cube datasets collected from real-world scenarios [40, 60]. For each radar

cube dataset, we measure the waveform parameters of the radar cube through formula fitting, ultimately selecting the average value of the results as the radar attribute label.

Synthetic dataset B. We draw inspiration from RadSimReal [3] to obtain the synthetic dataset B. Specifically, we generate multiple 3D reflection waveforms by presetting the dense waveform parameters, and then calculate the convolution of the reflection waveforms at each reflection point according to the method in RadSimReal, and finally obtain the synthetic radar cube. Overall, although synthetic dataset B is of lower quality than real dataset A, it covers a much richer set of radar attributes.

Simulation dataset C. To improve the overall quality of the mixed dataset, we generate the simulation dataset C using the original 3D U-Net (distinct from WARP-Net) without incorporating attribute embeddings. Specifically, we train a separate 3D U-Net on each real radar cube dataset, i.e., each model weight is dedicated to learning radar simulation under one radar attribute. Although not capable of capturing radar attributes, each trained 3D U-Net can generate accurately the radar cube for a specific radar attribute in different scenarios. We inference these 3D U-Nets on their respective unseen scenarios to generate a series of high-quality radar cubes. For example, assuming the existence of N real datasets, these correspond to N different radar attributes and N scene sets. By pairing them in all possible combinations, we can generate $N * (N - 1)$ new simulation datasets, significantly enriching our mixed dataset.

3.3.2. Training

We use L1 loss to train our WARP-Net. However, due to the random and dense distribution of the noisy reflection signals in all dimensions of the radar cube (Fig 3), the number of noisy reflection points in the reflection environment tensor \mathbf{E} is much larger than the scene reflection points. To ensure the accuracy of the scene reflection signal, we additionally compute the loss at all scene reflection point locations and name it L_{scene} . The total loss is as follows:

$$L = \|\mathbf{R}_{\text{sim}} - \mathbf{R}_{\text{gt}}\|_1 + L_{\text{scene}} \quad (4)$$

in which

$$L_{\text{scene}} = \|\mathbf{R}_{\text{sim}}[P_s] - \mathbf{R}_{\text{gt}}[P_s]\|_1 \quad (5)$$

where \mathbf{R}_{gt} is the ground truth radar cube and P_s denotes the coordinate index of all scene reflection points in the radar cube (and also in \mathbf{E}).

4. Experiments

4.1. Implementation Details

We perform our experiments on a single NVIDIA A800 GPU. For WARP-Net, we train the model for 50 epochs

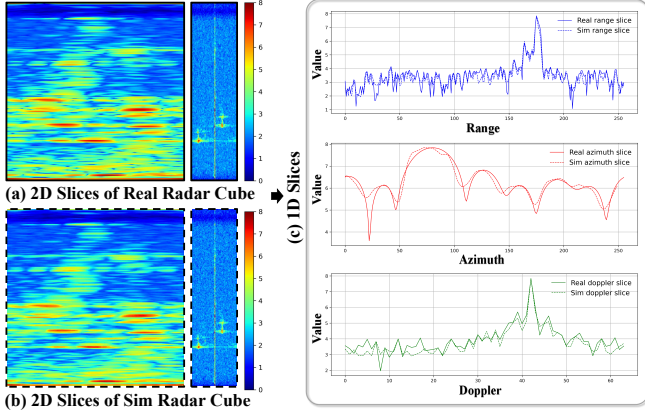


Figure 5. Comparison between the real radar cube and our simulation radar cube on RADDet [60].

Method	RADDet					Carrada				
	PPE	PPE _s	PPSE	FID	T _{sim}	PPE	PPE _s	PPSE	FID	T _{sim}
RadSimReal [3]	0.644	0.022	48.0	18.8	0.61s	0.634	0.063	48.9	45.2	0.65s
WARP-Net w/o L_{scene}	0.260	0.023	19.9	1.4	0.04s	0.261	0.038	20.0	20.0	0.04s
WARP-Net w/ L_{scene}	0.267	0.009	20.0	1.1	0.04s	0.271	0.012	20.1	12.6	0.04s

Table 1. Quantitative comparisons across different datasets. PPE and PPSE denote per-point error and per-point spectral error, respectively. The s denote reflected point locations. Our WARP-Net is trained on the proposed mixed dataset (completely isolated from the test sets). See Appendix § A.5 for calculation of above metrics.

with a batch size of 3, using the AdamW optimizer and a one-cycle learning rate schedule with a learning rate of 2×10^{-4} . All experimental datasets are collected from real-world road scenarios, with $r = 256$, $a = 256$, and $d = 64$. Training ($bs = 3$) and inference ($bs = 1$) require 20.5 GB and 2.5 GB of GPU memory, respectively. The training takes around one day. For a fair comparison, the models in each downstream experiment are trained using the same optimization strategy. See the Appendix for further details and additional experimental results, including the dataset description and ablation studies of training datasets, etc.

4.2. Analysis of Simulation Results

To verify the utility of the proposed Ctrl-RS, we evaluate the system in various aspects, including simulation quality, simulation efficiency, and attribute controllability. The following are the specific evaluation results.

Simulation Quality and Efficiency. To accurately assess the similarity between simulation and real data on the same scene, we extract all reflection points and reflection intensities (both scene and noise) from the real radar cube using the CFAR algorithm. We then simulate the radar cube using WARP-Net. Table 1 records the simulation error and time consumption of both methods on different datasets. Our method achieves per-point errors below 0.27 on both RADDet [60] and Carrada [40]. After introducing L_{scene} , although the global error increases slightly, the error of the

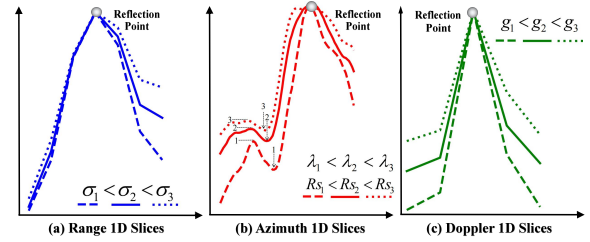


Figure 6. Comparison of 1D slices of the simulation results for the same reflection point with different waveform parameters. We intercept the part of the 1D slices that is affected by the labeled reflection point.

Test Set	Train Set	AP	AP@0.5	AP@0.75
RADDet	RADDet (R.)	24.9	47.4	22.3
	Sim-R-by-R + Sim-R-by-C	25.9	48.8	24.3
	R. + Sim-R-by-R	27.6	51.2	25.0
	R. + Sim-R-by-C	27.6	<u>52.2</u>	<u>25.2</u>
	R. + Sim-R-by-R + Sim-R-by-C	28.5_{+3.6}	54.7_{+7.3}	25.7_{+3.4}
Test Set	Train Set	AP	AP@0.5	AP@0.75
Carrada	Carrada (C.)	12.8	36.6	6.4
	Sim-C-by-C + Sim-C-by-R	24.0	49.8	19.6
	C. + Sim-C-by-C	21.7	48.6	15.7
	C. + Sim-C-by-R	22.7	<u>50.4</u>	16.3
	C. + Sim-C-by-C + Sim-C-by-R	29.1_{+16.3}	57.0_{+20.4}	26.4_{+20.0}

Table 2. 2D Object detection performance (AP) on RADDet and Carrada using RTMDet-Tiny [35] as baseline. See Fig 1 for bar chart. Red numbers represent improvements over the baseline (same applies to following tables).

scene reflection signal decreases significantly, which proves the effectiveness of the supervision. In addition, the global error of our method is significantly smaller compared to RadSimReal [3], as we accurately model the widely distributed noise reflection signals in the radar cube. Fig 5 shows the visualization results. Our simulation results are highly similar to the real radar cube, including the noise reflection signal. In terms of efficiency, our radar simulation takes around 0.04s, which is much faster than RadSimReal and traditional physical simulations [1, 20, 44, 59] (which generally take at least 5s). Additionally, we also record the full runtime of our method under different scene perception modes, as shown in Table 6. Overall, the majority of our method’s computational cost stems from environmental simulation, yet it can still be recognized as an efficient radar simulation approach.

Effectiveness of Attribute Embedding. In order to verify the ability of Ctrl-RS to capture different radar attributes, we compare the simulation results of the radar cube with different waveform parameters. As shown in Fig 6, our method can effectively sense the changes of radar attribute embedding and accurately align the waveforms in the radar cube. The simulation results under all radar attributes are geometrically similar to the real radar cube, which strongly demonstrates the physical reliability of our method.

Train Set	IoU (%)		Dice (%)	
	RA	RD	RA	RD
Carrada (C.)	39.2	54.4	48.5	66.3
Sim-C-by-C + Sim-C-by-R	37.0	57.9	44.9	70.7
C. + Sim-C-by-C	41.2	65.1	51.2	76.7
C. + Sim-C-by-R	39.6	63.4	49.1	75.0
C. + Sim-C-by-C + Sim-C-by-R	40.2	66.2	49.7	77.9

Table 3. Performance comparison of multi-view semantic segmentation on the Carrada test set. The baseline is MVRSS [39], which analyzes RA and RD views of continuously acquired radar signals to semantically segment them. See Fig 1 for bar chart. Complete results for each category are provided in Table A2 from Appendix.

4.3. Novel Sensor Trajectory

In this section, we explore Ctrl-RS’s ability to synthesize a radar cube in a new viewpoint. Fig 1 (a) shows the results of removing the target in the scene and moving the viewpoint 5 meters laterally (see the Appendix § A.8 for more results and implementation details). Our Ctrl-RS successfully reconstructs realistic radar cubes on new viewpoints and edited scenes, demonstrating its potential for generating new scenes in autonomous driving applications.

4.4. Downstream Validation

In this section, we explore the application value of Ctrl-RS simulation data through a variety of different downstream tasks. To exclude gains due to scenario diversity, we generate simulation data only on the training set scenarios of RADDet [60] and Carrada [40] (see Appendix § A.4 for a detailed description of these two datasets). For the radar attribute embedding, we set two sets of waveform parameter sets $\{\sigma, g, Rs, \lambda\}_R$ and $\{\sigma, g, Rs, \lambda\}_C$, which cover but are not limited to the radar attributes of RADDet and Carrada, respectively. Then, we generate four simulation datasets: **1) Sim-R-by-R**, which is generated on the RADDet training set scenario with the radar attribute from $\{\sigma, g, Rs, \lambda\}_R$. **2) Sim-R-by-C**, which is generated on the RADDet training set scenario with the radar attribute from $\{\sigma, g, Rs, \lambda\}_C$. **3) Sim-C-by-C**, which is generated on the Carrada training set scenario with the radar attribute comes from $\{\sigma, g, Rs, \lambda\}_C$. **4) Sim-C-by-R**, which is generated on the Carrada training set scenario with the radar attribute comes from $\{\sigma, g, Rs, \lambda\}_R$.

In the following, we show the experiment results on each task specifically (more experimental details are provided in the Appendix, along with comparisons to RadSimReal [3] and standard data augmentation).

2D Object Detection. First, we evaluate the effectiveness of the simulation data on the 2D object detection task. Using RTMDet-Tiny [35] as the baseline, we conduct five groups of experiments on the RADDet and Carrada datasets, under identical training settings for fair comparison. Objects are detected on Range-Doppler slices of the

3D radar cube, which are treated as 2D images. This differs from the 2D YOLO Head of RADDet, which performs 3D detection by outputting bounding boxes in the range-azimuth plane. The quantitative results are presented in Table 2. We observe that training solely on our simulation data already outperforms training on real data. Moreover, joint training with both simulation and real data achieves the best performance. These results demonstrate that Ctrl-RS’s simulation data can not only serve as a substitute for real data in 2D detection, but also enhance diversity and provide effective data augmentation.

Multi-View Radar Semantic Segmentation. In order to verify the semantic accuracy of the simulation data of Ctrl-RS, we conduct experiments on MVRSS [39]. MVRSS is a multi-view semantic segmentation method based on the radar cube, which analyzes RA (Range-Azimuth), RD (Range-Doppler) views of consecutive frames to semantically segment them. Same as other downstream tasks, we train MVRSS on different combinations of datasets and then directly compare their performance on a real test set. Table 3 illustrates the quantitative comparison results. The model trained entirely on our simulation data achieves better performance than the model trained on the real dataset. In addition, we achieve further improvements when using joint training with simulation data and real data. These results demonstrate not only the semantic accuracy of these simulation data, but also the stability of Ctrl-RS’s simulation on dynamic scenes.

3D Object Detection. We take the 3D radar object detection model RADDet as the baseline. Table 4 and Table 5 show the results under different training datasets. Unlike in the 2D detection task, 3D detection model trained on real radar data outperforms the model trained entirely on simulation radar data. Nevertheless, models trained using both real and simulation datasets still outperform models trained using only real datasets. These results reveal two important insights: (1) The 3D detection task imposes higher quality requirements on the simulation data because it involves finer 3D features, which are difficult to be fully captured by simulation methods (we suggest that this task should be considered as a challenge for radar simulation). (2) Our simulation data is proved to be valuable from a 3D perspective, which effectively enriches the diversity of radar attributes in the dataset and enhances the performance of existing models on complex tasks. See Fig A8 in Appendix for more visual results.

Furthermore, for both 2D and 3D detection, we use the simulator trained on Carrada to infer the RADDet scenarios, and the obtained simulation data still improves the model’s performance on the RADDet test set, i.e., “R. + Sim-R-by-C” outperforms “R.”. Swapping the two datasets, the above conclusion still holds. This further demonstrates that the performance gains stem from the diversity of radar at-

Test Set	Train Set	AP@0.3 of RAD / RA					
		Person	Bicycle	Car	Bus	Truck	All
RADDet	RADDet (R.)	32.91 / 51.80	22.25 / 39.88	65.42 / 81.20	43.42 / 42.11	51.55 / 62.63	55.50 / 70.91
	Sim-R-by-R + Sim-R-by-C	26.05 / 54.50	7.80 / 46.34	56.71 / 78.00	36.84 / 47.37	46.88 / 67.15	47.72 / 70.48
	R. + Sim-R-by-R	34.22 / 56.72	25.14 / 57.66	67.94 / 83.29	44.74 / 50.00	54.18 / 71.63	57.72 / 75.37
	R. + Sim-R-by-C	<u>36.25</u> / 63.68	30.64 / 57.37	<u>68.00</u> / 85.18	52.63 / 47.37	52.70 / 69.36	<u>58.25</u> / <u>77.50</u>
	R. + Sim-R-by-R + Sim-R-by-C	36.85 / <u>62.97</u>	<u>27.75</u> / 61.13	69.25 / 85.94	<u>44.74</u> / 60.53	57.64 / <u>70.38</u>	59.66 ^{+4.16} / 78.02 ^{+7.11}

Table 4. 3D Object detection AP@0.3 on RADDet. The baseline is the RADDet model [60], which takes as input a 3D radar cube and predicts 2D boxes via a RA YOLO Head and 3D boxes via a RAD YOLO Head. See Fig 1 for bar chart.

Test Set	Train Set	RAD	RA
Carrada	Carrada (C.)	9.83	29.91
	Sim-C-by-C + Sim-C-by-R	7.68	29.19
	C. + Sim-C-by-C	12.10	<u>31.39</u>
	C. + Sim-C-by-R	<u>13.05</u>	28.25
	C. + Sim-C-by-C + Sim-C-by-R	13.21 ^{+3.38}	31.55 ^{+1.64}

Table 5. 3D Object detection AP@0.3 on Carrada. The baseline is the RADDet model [60], which takes as input a 3D radar cube and predicts 2D boxes via a RA YOLO Head and 3D boxes via a RAD YOLO Head. See Fig 1 for bar chart. Complete quantitative results for each category are provided in Table A4 from Appendix.

Test Set	Train Set	AP@0.3	T _{all}
Sim-R _{test} -by-R	RADDet (R.)	58.16	0.30s
	R. + Sim-R-by-R + Sim-R-by-C	68.63	
Sim-N _{LiDAR} -by-R (*)	RADDet (R.)	52.14	0.64s
	R. + Sim-R-by-R + Sim-R-by-C	60.85	
Sim-N _{Camera} -by-R (*)	RADDet (R.)	48.96	1.29s
	R. + Sim-R-by-R + Sim-R-by-C	54.36	

Table 6. 3D Object detection AP@0.3 of RA boxes on simulation data. Sim-R_{test}-by-R: generated on RADDet test set scenes with the radar attribute from $\{\sigma, g, Rs, \lambda\}_R$. Sim-N-by-R: generated on nuScenes v1-mini [5] scenes with the radar attribute from $\{\sigma, g, Rs, \lambda\}_R$. We implement scene perception on nuScenes based on LiDAR and camera separately. T_{all}: runtime per scene for Ctrl-RS. *: more details and results for each category can be found in the Appendix § A.11.

tributes rather than overfitting to existing attributes.

4.5. Simulation based on Other Sensors

We validate the effectiveness of simulation data under different scene perception methods through downstream models. As shown in Table 6, even when employing diverse sensor data for scene perception, models trained solely on real-world data still achieve stable performance on our simulation data. We also evaluate the statistical similarity between these data simulation from other sensors and real radar data. The FID scores for simulation results from Lidar and camera relative to the RADDet test set are 35.8 and 47.1, respectively. These not only demonstrates the physical soundness of our simulation data but also validates the robustness of Ctrl-RS in selecting reflection points. More importantly, this conclusion is verified on an unseen scene

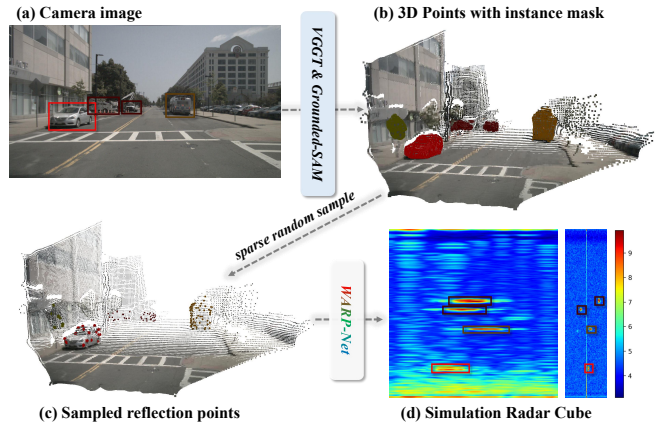


Figure 7. Radar simulation visualization based on nuScenes [5] camera data. The colored spheres in subfigure (c) represent the sampled points. See Fig A5 in Appendix for more examples.

(nuScenes [5]), providing crucial reference for the feasibility of Ctrl-RS in real-world scenarios. Fig 7 visualizes the complete process of how Ctrl-RS achieves radar simulation based on camera data.

Furthermore, in this experiment, the model trained using both real and simulation datasets consistently outperforms models trained only using real data, once again validating the practical value of our simulation data in real-world applications.

5. Conclusion

In this paper, we present Ctrl-RS, a module-complete radar data simulation method designed to simulate radar cubes under definable and controllable radar attributes for arbitrary scenarios. By integrating waveform parameter-based radar attribute embedding, Ctrl-RS can efficiently capture the variations of the radar cube under different radar attributes and accurately simulate radar data that match the real data distribution in range, azimuth and Doppler dimensions. Extensive experiments show that our approach significantly enhances the performance of existing models on various downstream tasks and demonstrate the potential of our approach as a generalized radar data engine for autonomous driving applications.

Limitations and Future Work. The waveform parameter-based representation of radar attributes in Ctrl-RS makes the control of radar attributes simple and intuitive, but it

also limits the perception of more detailed radar attributes, which makes it an incomplete substitute for real data in tasks with fine data distributions (e.g., 3D object detection). In addition, the lack of benchmarks makes it difficult to systematically and comprehensively compare our approach with existing methods. Naturally, we hope to address these issues in future work in order to contribute to the advancement of the radar simulation field.

Autonomous driving simulators still lack high-fidelity radar, even though radar is critical for robust perception in adverse weather. A key obstacle is that raw radar point clouds are extremely sparse and stochastic, making it difficult to model; we argue that simulating the full range–azimuth–Doppler cube is a more principled target. Existing radar cube simulators either rely purely on neural generators, which are opaque and offer little control over sensor attributes, or on detailed electromagnetic pipelines, which are slow, require proprietary hardware specifications, and still struggle to capture real-world complexity. We introduce Ctrl-RS, a controllable radar cube simulation framework that combines the strengths of both worlds. First, we build an environment reflection tensor from diverse sensor sources (including LiDAR, monocular cameras, and existing radar). Second, we abstract radar physics into a compact set of waveform parameters that characterize the 3D point spread function, yielding an intuitive embedding of radar attributes such as range resolution, Doppler broadening, and azimuth beam shape. Third, we train a WARP-Net on a large mixed dataset that fuses real, analytically synthesized, and simulator-generated radar cubes to cover a wide distribution of radar attributes. Ctrl-RS supports viewpoint changes, actor removal, and attribute editing. Experiments on RADDet, Carrada, and nuScenes show that our simulated data can match or surpass real radar in 2D detection and semantic segmentation, and consistently boosts performance in 3D detection when combined with real data. The Project is available at <https://github.com/zhuxing0/Ctrl-RS>.

References

- [1] Stefan Auer, Richard Bamler, and Peter Reinartz. Raysar-3d sar simulator: Now open source. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 6730–6733. IEEE, 2016. 2, 6, 1
- [2] Geonho Bang, Minjae Seong, Jisong Kim, Geunju Baek, Daye Oh, Junhyung Kim, Junho Koh, and Jun Won Choi. Rctdistill: Cross-modal knowledge distillation framework for radar-camera 3d object detection with temporal fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25315–25324, 2025. 1
- [3] Oded Bialer and Yuval Haitman. Radsimreal: Bridging the gap between synthetic and real data in radar object detection with simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15407–15416, 2024. 2, 3, 4, 5, 6, 7, 1
- [4] David Borts, Erich Liang, Tim Broedermann, Andrea Ramazzina, Stefanie Walz, Edoardo Palladin, Jipeng Sun, David Brueggemann, Christos Sakaridis, Luc Van Gool, et al. Radar fields: Frequency-space neural scene representations for fmcw radar. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. 2
- [5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 8, 3, 11
- [6] Carlo Capsoni and Michele D’Amico. A physically based radar simulator. *Journal of Atmospheric and Oceanic Technology*, 15(2):593–598, 1998. 2
- [7] Yujeong Chae, Heejun Park, Hyeonseong Kim, and Kuk-Jin Yoon. Doppler-aware lidar-radar fusion for weather-robust 3d detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27197–27208, 2025. 2
- [8] Xiaomeng Chu, Jiajun Deng, Guoliang You, Yifan Duan, Houqiang Li, and Yanyong Zhang. Racformer: Towards high-quality 3d object detection via query-based radar-camera fusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17081–17091, 2025. 1
- [9] Markus Clemens and Thomas Weiland. Discrete electromagnetism with the finite integration technique. *Progress In Electromagnetics Research*, 32(32):65–87, 2001. 2
- [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
- [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. 2
- [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. 1
- [13] Florian Engels, Philipp Heidenreich, Markus Wintermantel, Lukas Stäcker, Muhammed Al Kadi, and Abdelhak M Zoubir. Automotive radar signal processing: Research directions and practical challenges. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):865–878, 2021. 1, 2
- [14] Alfonso Farina and Francisco A Studer. A review of cfar detection techniques in radar systems. *Microwave Journal*, 29:115, 1986. 3
- [15] 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 genera-

- tive adversarial networks. *arXiv preprint arXiv:2308.02632*, 2023. 2
- [16] Demetrio Gubelli, Oleg A Krasnov, and Olexander Yarovy. Ray-tracing simulator for radar signals propagation in radar networks. In *2013 European Radar Conference*, pages 73–76. IEEE, 2013. 2, 4
- [17] John K Haas. A history of the unity game engine. 2014. 1
- [18] Yuval Haitman and Oded Bialer. Doppdrive: Doppler-driven temporal aggregation for improved radar object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26085–26094, 2025. 2
- [19] Danping He, Ke Guan, Bo Ai, Zhangdui Zhong, Junhyeong Kim, Heesang Chung, and Andrej Hrovat. Channel measurement and ray-tracing simulation for 77 ghz automotive radar. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):7746–7756, 2022. 2, 4
- [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, 6, 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*, pages 24118–24129, 2024. 2
- [22] Tianshu Huang, Akarsh Prabhakara, Chuhan Chen, Jay Karhade, Deva Ramanan, Matthew O’toole, and Anthony Rowe. Towards foundational models for single-chip radar. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24655–24665, 2025. 2
- [23] Xun Huang, Jinlong Wang, Qiming Xia, Siheng Chen, Bisheng Yang, Xin Li, Cheng Wang, and Chenglu Wen. V2x-r: Cooperative lidar-4d radar fusion with denoising diffusion for 3d object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27390–27400, 2025. 3
- [24] Jian-Ming Jin. *The finite element method in electromagnetics*. John Wiley & Sons, 2015. 2
- [25] Woo-Jin Jung, Dong-Hee Paek, and Seung-Hyun Kong. 4d radar ground truth augmentation with lidar-to-4d radar data synthesis. *arXiv preprint arXiv:2503.03637*, 2025. 2
- [26] Pou-Chun Kung, Skanda Harisha, Ram Vasudevan, Aline Eid, and Katherine A Skinner. Radarsplat: Radar gaussian splatting for high-fidelity data synthesis and 3d reconstruction of autonomous driving scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27596–27606, 2025. 1
- [27] Haowen Lai, Zitong Lan, and Mingmin Zhao. Non-line-of-sight 3d reconstruction with radar. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2
- [28] Zhengxin Lei, Feng Xu, Jiangtao Wei, Feng Cai, Feng Wang, and Ya-Qiu Jin. Sar-nerf: Neural radiance fields for synthetic aperture radar multi-view representation. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 2
- [29] Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. *arXiv preprint arXiv:2412.05435*, 2024. 1
- [30] Jian Li and Petre Stoica. *MIMO radar signal processing*. John Wiley & Sons, 2008. 1, 2
- [31] Qian Liang, Ruixu Geng, Jinbo Chen, Haoyu Wang, Yan Chen, and Yang Hu. Spatial alignment and temporal matching adapter for video-radar remote physiological measurement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8623–8633, 2025. 3
- [32] Yunfei Long, Abhinav Kumar, Xiaoming Liu, and Daniel Morris. Riccardo: Radar hit prediction and convolution for camera-radar 3d object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22276–22285, 2025. 3
- [33] John Ludbrook. Comparing methods of measurement. *Clinical and Experimental Pharmacology and Physiology*, 24(2): 198–203, 1997. 5
- [34] Van-Tin Luu, Yon-Lin Cai, Vu-Hoang Tran, Wei-Chen Chiu, Yi-Ting Chen, and Ching-Chun Huang. Rc-autocalib: An end-to-end radar-camera automatic calibration network. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6700–6709, 2025. 3
- [35] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. RtmDET: An empirical study of designing real-time object detectors, 2022. 6, 7, 5
- [36] Takashi Machida and Takashi Owaki. Rapid and precise millimeter-wave radar simulation for adas virtual assessment. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 431–436. IEEE, 2019. 2
- [37] Bassem R Mahafza and Atef Elsherbeni. *MATLAB simulations for radar systems design*. Chapman and Hall/CRC, 2003. 3
- [38] Michael Meyer, Georg Kuschik, 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. 2
- [39] Arthur Ouaknine, Alasdair Newson, Patrick Pérez, Florence Tupin, and Julien Rebut. Multi-view radar semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15671–15680, 2021. 7, 6
- [40] 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. 5, 6, 7, 3, 4
- [41] Sujeet Milind Patole, Murat Torlak, Dan Wang, and Murataza Ali. Automotive radars: A review of signal processing techniques. *IEEE Signal Processing Magazine*, 34(2):22–35, 2017. 1
- [42] Kainat Yasmeen Shobha Sundar Ram. Synthesis of through-wall micro-doppler signatures of human motions using generative adversarial networks. *arXiv preprint arXiv:2404.08739*, 2024. 2
- [43] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng

- Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 3
- [44] Christian Schüßler, Marcel Hoffmann, Johanna Bräunig, Ingrid Ullmann, Randolph Ebel, 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, 6, 1
- [45] Mehrdad Soumekh. *Synthetic aperture radar signal processing*. New York: Wiley, 1999. 1, 2
- [46] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14895–14904, 2024. 1
- [47] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vgg: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 3
- [48] Leichen Wang, Bastian Goldluecke, and Carsten Anklam. L2r gan: Lidar-to-radar translation. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2
- [49] Shengpeng Wang, Yulong Xie, Qing Liao, and Wei Wang. S3e: Self-supervised state estimation for radar-inertial system. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26686–26695, 2025. 2
- [50] 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. 4, 2
- [51] Yiran Wang, Jiaqi Li, Chaoyi Hong, Ruibo Li, Liusheng Sun, Xiao Song, Zhe Wang, Zhiguo Cao, and Guosheng Lin. Tacodepth: Towards efficient radar-camera depth estimation with one-stage fusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10523–10533, 2025. 1
- [52] 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
- [53] 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
- [54] Jialong Wu, Marco Braun, Dominic Spata, and Matthias Rottmann. Tars: Traffic-aware radar scene flow estimation. *arXiv preprint arXiv:2503.10210*, 2025. 2
- [55] Yuyan Wu and Hae Young Noh. Road boundary detection using 4d mmwave radar for autonomous driving. *arXiv preprint arXiv:2503.01930*, 2025. 1
- [56] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pages 156–173. Springer, 2024. 1
- [57] Lei Yang, Xinyu Zhang, Jun Li, Chen Wang, Jiaqi Ma, Zhiying Song, Tong Zhao, Ziyang Song, Li Wang, Mo Zhou, et al. V2x-radar: A multi-modal dataset with 4d radar for cooperative perception. *arXiv preprint arXiv:2411.10962*, 2024. 2
- [58] Kane Yee. Numerical solution of initial boundary value problems involving maxwell’s equations in isotropic media. *IEEE Transactions on antennas and propagation*, 14(3):302–307, 1966. 2
- [59] Zhengqing Yun and Magdy F Iskander. Ray tracing for radio propagation modeling: Principles and applications. *IEEE access*, 3:1089–1100, 2015. 2, 6, 1
- [60] Ao Zhang, Farzan Erlik Nowruzi, and Robert Laganieri. 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. 4, 5, 6, 7, 8, 1, 2
- [61] Liwen Zhang, Xinyan Zhang, Youcheng Zhang, Yufei Guo, Yuanpei Chen, Xuhui Huang, and Zhe Ma. Peakconv: Learning peak receptive field for radar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17577–17586, 2023. 4, 6
- [62] Hanzhi Zhong, Zhiyu Xiang, Ruoyu Xu, Jingyun Fu, Peng Xu, Shaohong Wang, Zhihao Yang, Tianyu Pu, and Eryun Liu. Cvfusion: Cross-view fusion of 4d radar and camera for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28188–28197, 2025. 1
- [63] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21634–21643, 2024. 1

Controllable Radar Simulation with Waveform Parameter Embedding

Supplementary Material

In the Supplementary Material, we provide the Appendix (Sec A) of this paper along with the project homepage and reproduction code. For additional visualization results and code details, please refer to the **ACCOMPANYING FILES** in the folder.

A. Appendix

The supplementary material herein extends the discussion and analysis presented in the primary manuscript. It is structured as follows:

Introduction to RadSimReal (§ A.1) : This section introduces the RadSimReal [3] method and discusses its key details.

Standard 3D Reflection Signals and Fitting Equations Across Different Datasets (§ A.2) : This section shows the geometric consistency of the standard 3D reflection signal across different dimensions in various datasets, and describes how we formulaically model the waveforms of each dimension of the standard 3D reflection signal.

Network Structure of WARP-Net (§ A.3) : This section describes in detail the network architecture of WARP-Net.

Datasets (§ A.4) : This section describes the RADDet and Carrada datasets that we used in WARP-Net’s training and the downstream tasks.

Formula for Calculating Radar Simulation Quantitative Metrics (§ A.5): This section details the calculation formulas for the metrics in Table 1, which are used to assess the similarity between the simulation radar cube and the real radar cube.

Mixed Dataset for WARP-Net’s Training and Ablation Study (§ A.6) : This section describes the implementation details of the hybrid dataset used to train WARP-Net, and explores the impact of different training data on the simulation performance of WARP-Net through ablation study.

More Visualization of the Simulation Data (§ A.7) : This section shows more simulation results for the Ctrl-RS.

Details of Novel Sensor Trajectory (§ A.8) : This section describes the implementation details of editing the scene and radar attributes using Ctrl-RS.

Training settings and more results for different downstream tasks (§ A.9): This section describes the training setup for multiple downstream tasks and additional results.

Comparison with RadSimReal and Standard Data Augmentation for Downstream Task (§ A.10) : This section compares the performance of Ctrl-RS with RadSimReal and standard data augmentation for 3D target detection, validating the quality of our simulation data.

Additional experimental results on unseen sensor and scenarios (§ A.11) : In this section, we proceed to re-evaluate the results presented in Table 6 of the main paper, focusing on the range-azimuth dimension to further analyze the performance of Ctrl-RS on unseen sensors and scenes.

A.1. Introduction to RadSimReal [3]

Physical radar simulation [1, 20, 44, 59] is a technique for generating a synthetic radar cube (range-azimuth-Doppler) by physically modeling the environment and radar sensor. The process usually consists of several steps: first, a scene (e.g., a road scene) is generated by a 3D modeling tool (e.g., Unity [17] or CARLA [12]); then, ray tracing techniques are used to compute the reflections of electromagnetic waves emitted from the radar on objects in the environment; and finally, radar-specific signal processing algorithms are applied to generate the radar cube. The key to this approach is an in-depth understanding of the radar hardware parameters and signal processing algorithms, which makes the simulation process very complex and computationally resource intensive.

Unlike traditional physical radar simulation, RadSimReal realizes radar cube generation via Point Spread Function (PSF).The core idea of RadSimReal is to use PSF to describe the radar sensor’s response to reflecting points without the need to deeply understand the radar’s specific design details and signal processing process. Specifically, users are able to extract the PSF from a simple set of radar measurements, and integrate the PSF to respond at each reflection point in the scene to quickly generate a high-quality synthetic radar cube (as shown in Equation (1) in the main paper).

For experimental validation, the researcher used RadSimReal to generate synthetic radar cubes containing annotations, and 2D slices of these cubes were used to train a 2D Object Detection model [60]. The experimental results show that the detection performance of the 2D Object Detection model trained using the data generated by RadSimReal is comparable to that of the model trained using only real data.

Overall, RadSimReal is inspiring in that modeling radar signals by PSF (generalized in this paper to standard 3D reflection signals) is feasible and can be used for downstream tasks. Based on this, we propose Ctrl-RS, which improves the accuracy and efficiency of radar simulation and realizes the controllability of radar attributes. Meanwhile, we validate the effectiveness of our simulation data on more complex tasks such as 3D Object Detection.

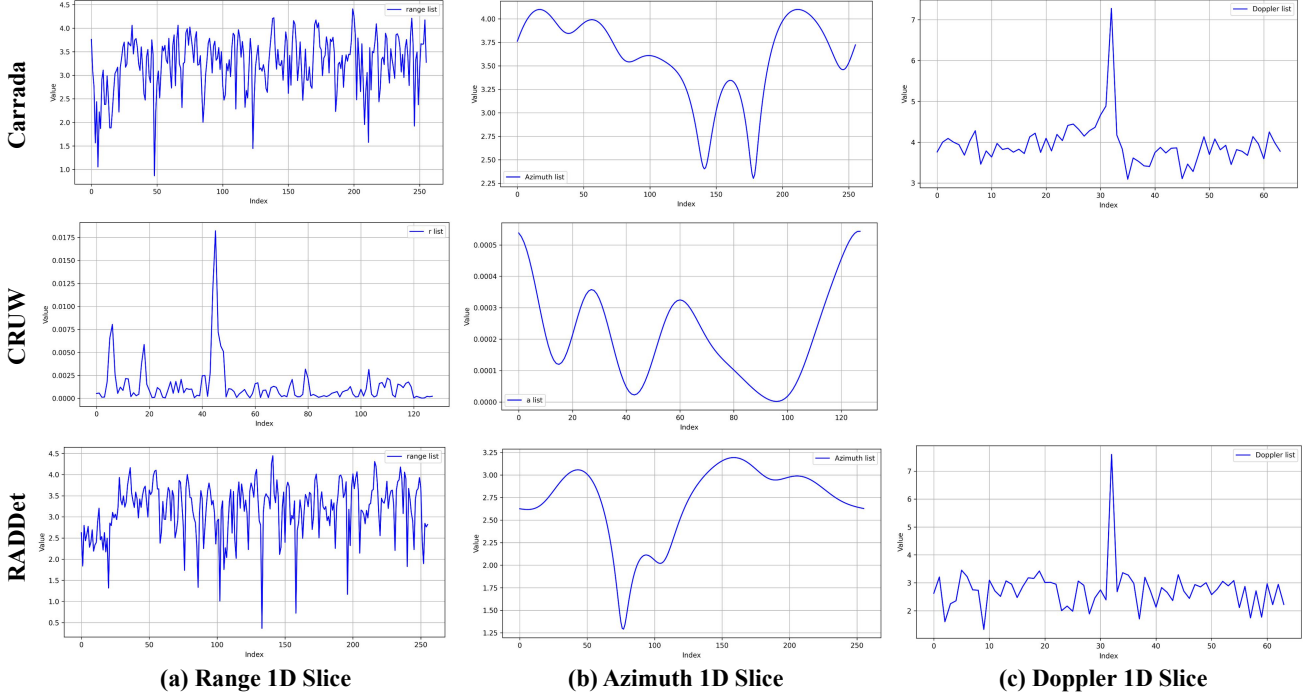


Figure A1. Radar cube 1D slices on different datasets.

A.2. Standard 3D Reflection Signals and Fitting Equations Across Different Datasets

We show 1D slices of the radar cube on the different datasets in Fig A1. Since the CRUW dataset [50] only provides RA data, we could not get its Doppler slices. Despite the different distributions of the values of the radar cube on the different datasets (because of the different treatments), the standard 3D reflectance signals of each dataset are geometrically very similar in the Range, Azimuth, and Doppler slices, which inspired us to use functions to fit them separately.

The specific formulas for fitting functions in different dimensions are as follows:

$$\mathbf{S}_R(r) = e^{-\frac{(r-r_i)^2}{2\sigma^2}} \quad (\text{A1})$$

$$\mathbf{S}_A(a) = |FFT((1-p) - p * \cos(\frac{2\pi n}{N-1}))| \otimes \delta(a - a_i) \quad (\text{A2})$$

$$\mathbf{S}_D(d) = g * \max\{1 - |d - d_i|, 2 - 4|d - d_i|, 0\} \quad (\text{A3})$$

where (r_i, d_i, a_i) is the coordinate index of the reflection point in the radar cube, σ denotes the standard deviation of the Gaussian function, FFT denotes Fast Fourier Transform, N and p are the length and parameters of the window function, \otimes denotes the convolution operation, and g is the gradient in the segmented linear function. With the above equation, we can further calculate the waveform parameter $\{\sigma, g, R_s, \lambda\}$ set in this paper to visually characterize the radar attribute.

A.3. Network Structure of WARP-Net

We propose WARP-Net, which directly simulates the radar cube based on the reflection environment tensor \mathbf{E} and the waveform parameter-based radar attribute. Fig A2 illustrates the architecture of WARP-Net. In order to increase the receptive field in different dimensions (especially the azimuth dimension), we construct the WARP-Net by four downsampling blocks and four upsampling blocks. Each downsampling block consists of two 3D convolutions with output channel numbers of 64, 128, 192, and 256, respectively. The first three upsampling blocks consist of one 3D transpose convolution and three 3D convolutions with output channel numbers of 192, 128, and 64, respectively, while the last upsampling block contains only one 3D transpose convolution and outputs the radar feature map with channel number of 8. In addition, for each 3D convolution and 3D transposed convolution in the WARP-Net, we process its output with a BN layer and a LeakyReLU layer. Finally, we decode the radar feature map using a simple $3 \times 3 \times 3$ 3D convolution and the Relu function to obtain the simulated radar cube $\mathbf{R}_{\text{sim}} \in \mathbb{R}^{r \times d \times a}$.

A.4. Datasets

RADDet [60]. The RADDet dataset is a publicly available radar dataset intended to be used for multi-class object detection by dynamic road users, and contains radar data denoted as range-azimuth-Doppler tensor and corresponding annotations covering multiple object classes (person, bicy-

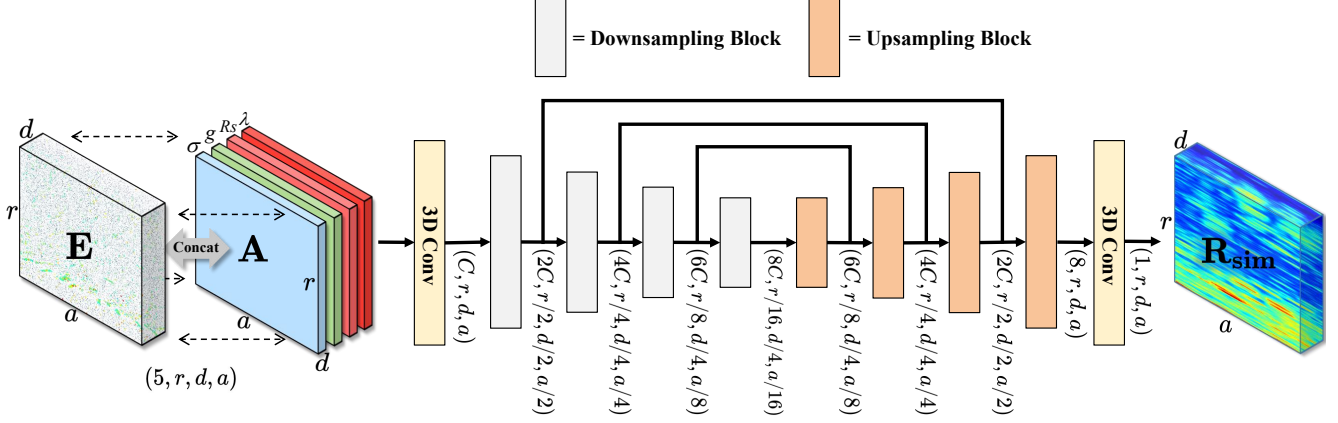


Figure A2. The Architecture of WARP-Net.

cle, car, motorcycle, bus, truck). The dataset was collected via a tat and a pair of stereo cameras, and data capture was performed under clear sky conditions. It contains 10,158 frames of radar data in 3D tensor (256, 256, 64) format containing rich dynamic object information. The training and test sets contain 8,127 and 2,031 frames of radar data, respectively. It should be noted that due to the insufficient number of motorcycle samples, we skip its quantitative assessment, but still consider it across all categories.

Carrada [40]. Carrada is an automotive radar and camera dataset that provides 12666 frames of data in 30 sequences covering a wide range of scenarios for pedestrians, cars, and bicyclists. The data consists of three forms of annotations: sparse dots, bounding boxes and dense masks, which are suitable for object detection, semantic segmentation and tracking tasks. The training set and test set contain 5,802 and 1,391 frames of usable radar data in 3D tensor (256, 256, 64) format, respectively. Carrada employs a semi-automatic annotation method based on camera information in order to improve efficiency and reduce cost, which also makes its annotation quality poor. We use its training set and validation set together for model training.

nuScenes [5]. nuScenes is a large-scale autonomous driving dataset. It comprises 1,000 carefully selected driving scenarios, each lasting 20 seconds, totaling approximately 15 hours of driving data. The dataset covers complex traffic environments in Boston and Singapore. The dataset includes approximately 1.4 million camera images, 390,000 LiDAR scans, 1.4 million radar scans, and 1.4 million object bounding boxes across 40,000 keyframes, suitable for training and testing autonomous driving algorithms. However, since radar scans are provided as sparse point clouds, which filter out weak reflections and noise in the scene, Therefore, we only utilize the Front Camera Images and Lidar Data provided in this dataset for radar simulation. Additionally, due to dataset size constraints, we utilize a subset of this dataset (i.e., v1-mini), containing approximately 400

frames of data.

A.5. Formula for Calculating Radar Simulation Quantitative Metrics

In Table 5 of the main text, we employed multiple metrics to evaluate the accuracy of the simulated data. Notably, unlike image data, our simulation data $\mathbf{R}_{\text{sim}} \in \mathbb{R}^{r \times d \times a}$ constitutes a three-dimensional tensor. Therefore, we apply dimensionality reduction during the computation of FID. This section details their respective calculation formulas.

PPE, or Per-Pixel Error, directly accounts for the average numerical discrepancy between simulation data and real data, calculated as follows:

$$\text{PPE} = E(|\mathbf{R}_{\text{sim}} - \mathbf{R}_{\text{gt}}|) \quad (\text{A4})$$

where $E(\cdot)$ denotes the average value of all the elements in it, $|\cdot|$ denotes the absolute value calculation, \mathbf{R}_{sim} is the simulated radar cube, and \mathbf{R}_{gt} is the ground truth radar cube.

PPSE, or Per-Point Spectral Error, evaluates the average error between simulation data and real data from the frequency domain perspective. Its calculation formula is as follows:

$$\text{PPSE} = E(|\text{FFT}(\mathbf{R}_{\text{sim}}) - \text{FFT}(\mathbf{R}_{\text{gt}})|) \quad (\text{A5})$$

where FFT denotes the Fourier Transform.

FID is a common metric used to assess statistical similarity between images. Therefore, we first reduce the dimensionality of the radar cube by taking the maximum Doppler dimension to obtain the range-azimuth image, as follows:

$$\mathbf{I}_{\text{sim}} = \max(\mathbf{R}_{\text{sim}}, \text{axis} = 1) \quad (\text{A6})$$

$$\mathbf{I}_{\text{gt}} = \max(\mathbf{R}_{\text{gt}}, \text{axis} = 1) \quad (\text{A7})$$

where \mathbf{I}_{sim} and \mathbf{I}_{gt} are radar images obtained by dimensionality reduction from \mathbf{R}_{sim} and \mathbf{R}_{gt} , respectively. We then

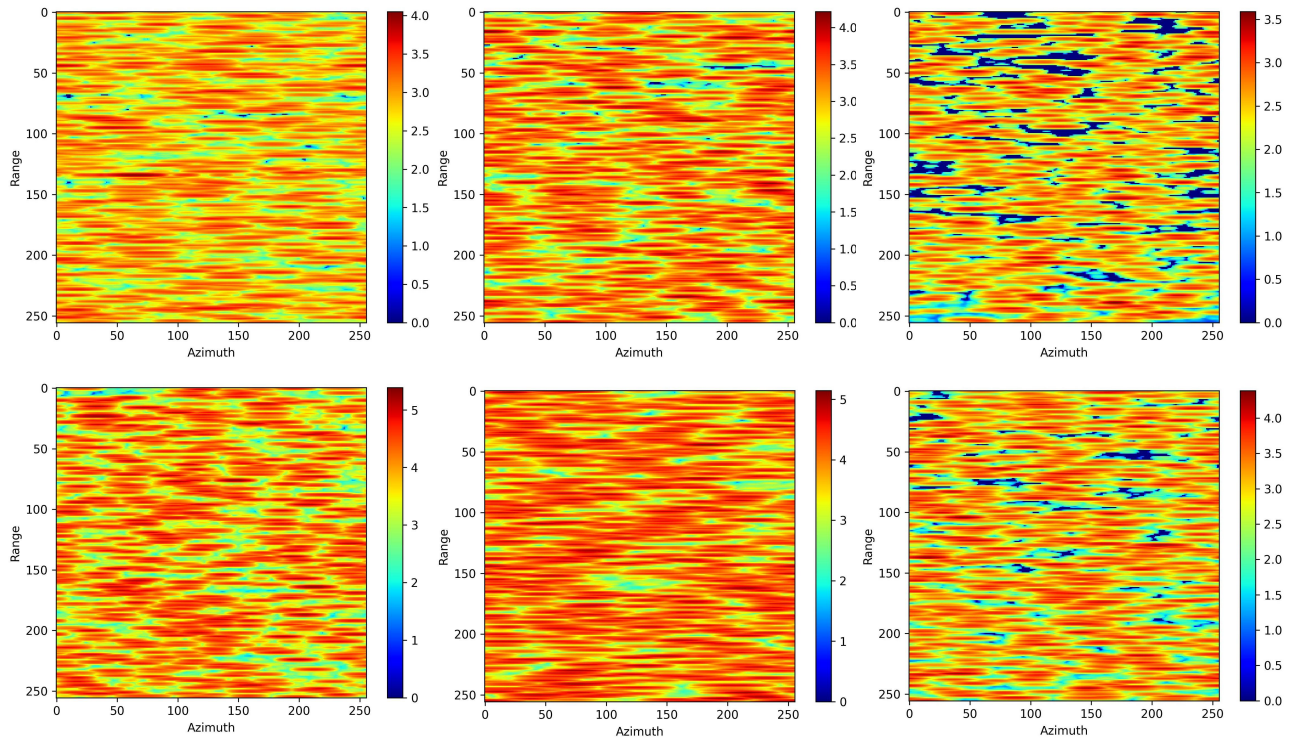


Figure A3. Simulation results of different standard 3D reflection signals at random noise reflection points.

compute their FID metrics as follows:

$$\text{FID} = \|\mu_{\text{sim}} - \mu_{\text{gt}}\|^2 + \text{Tr}(\Sigma_{\text{sim}} + \Sigma_{\text{gt}} - 2\sqrt{\Sigma_{\text{sim}}\Sigma_{\text{gt}}}) \quad (\text{A8})$$

where $\mu_{(\cdot)}$ denotes the mean vector of features for the radar images, $\Sigma_{(\cdot)}$ denotes the feature covariance matrix for the radar images, and $\text{Tr}(\cdot)$ denotes the trace of the matrix.

A.6. Mixed Dataset for WARP-Net’s Training and Ablation Study

Implementation Details. In this paper, we produce a mixed dataset rich in radar attributes to train WARP-Net. As we mentioned in the main paper, the mixed dataset is divided into three parts: the real dataset A, the synthetic dataset B, and the simulation dataset C. The details of the practice are described below.

Since there are fewer real radar cube datasets in the form of range-azimuth-Doppler tensor, we only use the training sets of RADDet [60] and Carrada [40] in practice to compose the real dataset A. Therefore, the real dataset A covers only two radar attributes.

For the production of the synthetic dataset B, we combine the method in RadSimReal [3] with the three fitting functions demonstrated in Sec A.2. We use the method in RadSimReal to produce the synthetic dataset B. The synthetic dataset B is a set of the three fitting functions.

Specifically, we construct the set of fitting functions by randomly selecting the fitting parameters from the following set: $\sigma \in \{2.4, 2.5, 2.6, 2.7, 2.8\}$, $N \in \{6, 7, 8, 9, 10\}$, $g \in \{0.5, 0.6, 0.7\}$ and $p \in \{0.1, 0.2, 0.3\}$. Note that we measure the average of the fitting parameters in the RADDet training set, which are: $\sigma = 2.6$, $N = 8$, $g = 0.6$ and $p = 0.1$. We then use these fitting functions to compute the standard 3D reflection signals according to Eq. (2) in the main paper (Fig A3 illustrates these signals on the random noise reflection points). Finally, we generate the radar data under these standard 3D reflection signals (randomly selected standard 3D reflection signals) in the scenario of the RADDet training set according to Eq. (1) in the main paper. To summarize, the number of samples in the synthetic dataset B is equal to the number of scenes in the RADDet training set, but the radar attributes are much richer.

For simulation dataset C, due to the lack of real datasets, we only train two 3D U-Nets with radar attribute embeddings removed on RADDet and Carrada. We use the two 3D U-Nets (with radar attribute embeddings removed) trained on RADDet and Carrada to inference on each other’s training set scenarios separately, and we finally obtain a simulated dataset C, which has the same number of samples as the real dataset.

Dataset A	Dataset B	Dataset C	RADDet [60]		Carrada [40]	
			PPE	PPE _s	PPE	PPE _s
✓			<u>0.271</u>	0.008	0.270	0.009
	✓		0.480	0.138	0.505	0.171
		✓	0.363	0.033	0.487	0.027
✓	✓	✓	0.267	<u>0.009</u>	<u>0.271</u>	<u>0.012</u>

Table A1. Comparison of WARP-Net simulation quality under different types of training data. **Bold**: Best. Underline: Second.

Finally, since the ratio of train sets between RADDet and Carrada is approximately 4: 3, the overall ratio of datasets A, B and C is roughly 7: 4: 7.

Ablation Study. We explore the effect of training data type on the performance of WARP-Net. As we do in the main paper, we extract all reflection points and reflection intensities (both scene and noise) from a real radar cube using the CFAR algorithm. Then, we simulate the radar cube using WARP-Net. Table A1 records the simulation errors of WARP-Net with different training data. Considering only the simulation quality, WARP-Net trained only with the real dataset A is slightly better than WARP-Net trained only with the simulation dataset C, and much better than WARP-Net trained only with the synthetic dataset B.

These results reveal two important insights: 1. The real dataset A significantly improves the simulation quality of WARP-Net. 2. Although the synthetic dataset B effectively enriches the radar attributes of the mixed dataset, it damages the simulation quality of WARP-Net (thus we use the simulation dataset C to mitigate this negative effect and further enrich the diversity of the mixed dataset).

A.7. More Visualization of the Simulation Data

Fig A4 presents additional simulation results of Ctrl-RS under the same conditions as Fig 5 in the main text. It can be observed that our simulation data exhibits high similarity to real-world data.

Furthermore, we present additional radar simulation results based on camera images in Fig A5 as a supplement to Fig 7 in the main text. Objectively speaking, Ctrl-RS’s results are inevitably influenced by scene reconstruction quality. For instance, inaccuracies in target positions within the 3D point cloud may introduce errors in simulation results. Nevertheless, our simulations consistently yield visually plausible outcomes and remain directly usable for downstream tasks, fully demonstrating Ctrl-RS’s robustness.

A.8. Details of Novel Sensor Trajectory

Our Ctrl-RS allows editing of the scene and editing of the simulation radar attributes by modifying the reflection environment tensor \mathbf{E} and radar attribute embedding. The visualization results are shown in Fig A6. The following are the

specific implementation details of each editing operation.

Modify Attribute. No need for complicated operation, we can directly modify the waveform parameters $\{\sigma, g, Rs, \lambda\}$ which are input to WARP-Net to simulate different radar cube with different radar attributes for the same scene.

Novel Trajectory. As described in Sec 3.1, the reflection environment tensor \mathbf{E} in Ctrl-RS corresponds dimensionally to the simulated radar cube. In other words, each of the three dimensions of the reflection environment tensor \mathbf{E} has a specific physical meaning, i.e., range, azimuth, and Doppler. Therefore, we are free to edit the viewpoint in the scene perception. Then, the new reflection environment tensor E_{new} is obtained by recalculating the corresponding coordinates of the scene reflection points in the reflection environment tensor under the new viewpoint. Finally, we use WARP-Net to process the reflective environment tensor \mathbf{E}_{new} to obtain the simulated radar cube under the new viewpoint.

Actor Removal. We realize the removal function by directly modifying the reflection intensity of the reflection points on the actor. Specifically, for the actor to be removed in the scene, we can box its reflection points by 3D Object detection, and then reduce the reflection intensity of its reflection points to be similar to that of the noise reflection points. Then, we directly use WARP-Net to process the intensity-modified reflection environment tensor to get the simulated radar cube of actor removal.

A.9. Training settings and more results for different downstream tasks

Multi-View Radar Semantic Segmentation. In this paper, we use MVRSS to verify the semantic accuracy and stability of Ctrl-RS simulation data. We present the complete results on the Carrada test set in Table A2, as a supplement to Table 3 in the main text. For model training, we only modified the number of training epochs in the default training configuration: from training 300 epochs to training 200 epochs, thus reducing the time cost. Nevertheless, the model trained jointly on our simulation data and real data still outperforms the model trained with 300 epochs in the original paper.

2D Object Detection. We conduct experiments on the RADDet and Carrada datasets for 2D object detection using Range-Doppler (RD) slices. For our object detection model, we adopt RTMDet-Tiny [35], which is trained on a single NVIDIA A800 GPU. Specifically, The model is trained for 100 epochs on the RADDet dataset and 300 epochs on the Carrada dataset, using the AdamW optimizer with a cosine annealing learning rate schedule. We initialize our model with COCO pre-trained weights and evaluate performance on annotated RD slices. Table A3 presents the number of RD slices in the training and test sets for each dataset. Fi-

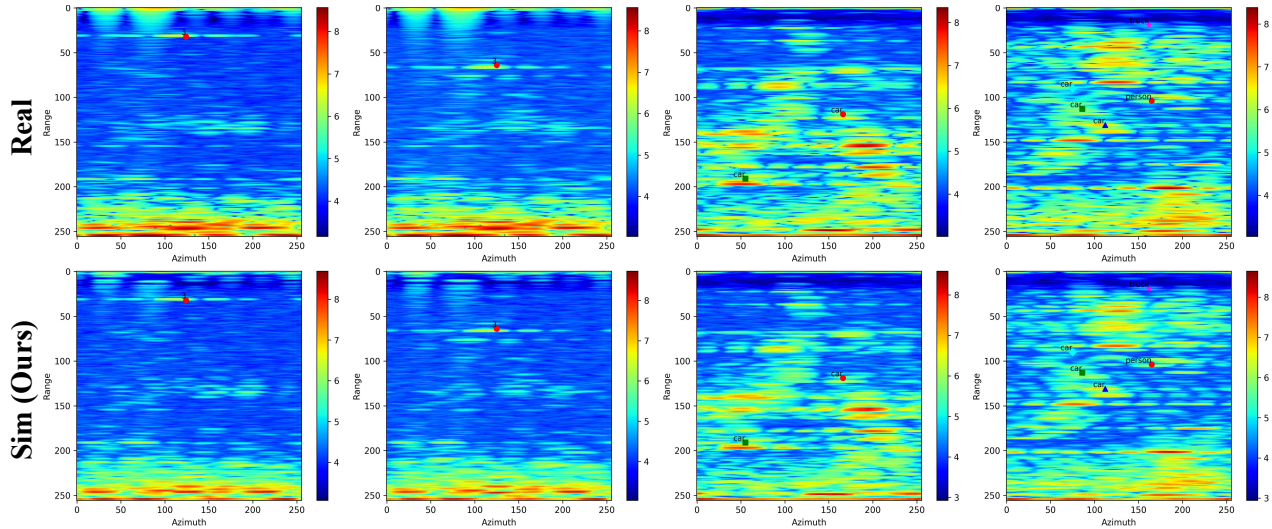


Figure A4. Simulation results on the RADDet and Carrada test sets.

Train Set	IoU (%) of RA / RD					Dice (%) of RA / RD				
	Bkg.	Ped.	Cyclist	Car	All	Bkg.	Ped.	Cyclist	Car	All
Carrada (C.)	99.7 / 99.7	12.1 / 23.7	15.7 / 38.1	29.2 / 56.3	39.2 / 54.4	99.8 / 99.8	21.6 / 38.3	27.1 / 55.1	45.2 / 72.0	48.5 / 66.3
Sim-C-by-C + Sim-C-by-R	99.8 / 99.7	8.4 / 46.3	10.3 / 38.9	29.3 / 46.7	37.0 / 57.9	99.9 / 99.8	15.5 / 63.3	18.8 / 56.0	45.3 / 63.7	44.9 / 70.7
C. + Sim-C-by-C	99.8 / 99.8	18.7 / 52.4	14.5 / 40.6	31.8 / 67.5	41.2 / 65.1	99.9 / 99.9	31.5 / 68.8	25.3 / 57.7	48.3 / 80.6	51.2 / 76.7
C. + Sim-C-by-R	99.8 / 99.8	14.7 / 54.1	14.8 / 33.8	29.2 / 66.0	39.6 / 63.4	99.9 / 99.9	25.6 / 70.2	25.8 / 50.5	45.2 / 79.5	49.1 / 75.0
C. + Sim-C-by-C + Sim-C-by-R	99.8 / 99.8	14.5 / 54.9	14.6 / 44.4	33.4 / 64.7	40.2 / 66.2	99.9 / 99.9	25.7 / 71.3	25.4 / 61.6	51.4 / 76.1	49.7 / 77.9

Table A2. Performance comparison of multi-view semantic segmentation on the Carrada test set. The baseline is MVRSS [39], which analyzes RA and RD views of continuously acquired radar signals to semantically segment them.

Train Set	Amounts	Train Set	Amounts
RADDet (R.)	22694	Carrada (C.)	5802
Sim-R-by-R + Sim-R-by-C	45388	Sim-C-by-C + Sim-C-by-R	11604
R. + Sim-R-by-R	45388	C. + Sim-C-by-C	11604
R. + Sim-R-by-C	45388	C. + Sim-C-by-R	11604
R. + Sim-R-by-R + Sim-R-by-C	68082	C. + Sim-C-by-C + Sim-C-by-R	17406
Test Set	Amount	Test Set	Amount
RADDet (R.)	5619	Carrada (C.)	1399

Table A3. The amount of RD slices in the training and test sets for the RADDet and Carrada datasets. For the RADDet dataset, we extract Range-Doppler (RD) slices by selecting the maximum value along the angle dimension. For the Carrada dataset, we determine the RD slices based on the bounding box dimensions of the "car" class, selecting the maximum value along the angle dimension accordingly.

nally, we select the best-performing models and evaluate their performance using bbox AP@0.5:0.95 (AP), AP@0.5, and AP@0.75.

For dataset selection, we adopt different strategies based on dataset characteristics. In the RADDet dataset, following 3D detection approaches, we select five representative object categories, including car, bicycle, person, bus and truck. In contrast, for Carrada, due to its poor annotation

quality, we regenerate new annotations [61] for each scene and focus on the representative car category for detection.

Fig A7 provides a comparison of some of the 2D detection results on the Carrada. Our simulation data from Ctrl-RS can effectively enhance the detection performance of the model.

3D Object Detection. In this paper, we use the RADDet model as a baseline in the 3D detection task. We follow its default training settings. Specifically, we train the model with a batch size of 3 for 1000 epochs using the Adam optimizer. The initial learning rate is 0.0001; after 60K warm-up steps, the learning rate decays to 0.96 every 10K steps. The NMS thresholds for the 3D and 2D detection headers are 0.1 and 0.3, respectively. In fact, we find that the model has largely converged or even started to overfit after being trained for about 300 epochs, so we save the best-performing weights for comparison.

Similarly, we present the complete results on the Carrada test set in Table A4, as a supplement to Table 5 in the main text. Nevertheless, the visualization results of the experiments are shown in Fig A8.

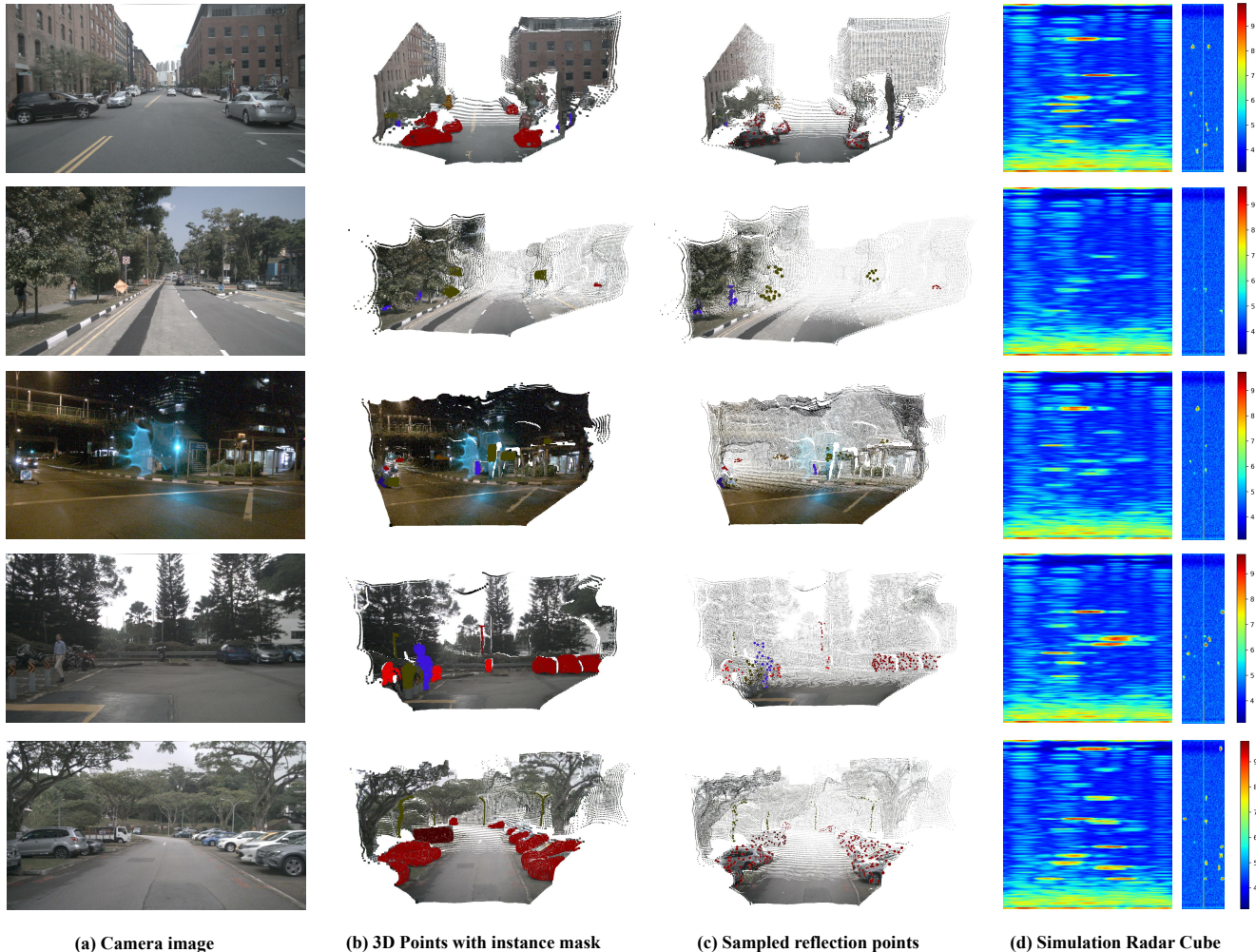


Figure A5. Radar simulation results based on camera images. Note that since the dataset does not provide precise speed information, we randomly assigned speeds to different instances in the simulation, which does not affect the validation on downstream tasks.

Test Set	Train Set	AP@0.3 of RAD / RA			
		Pedestrian	Cyclist	Car	All
Carrada	Carrada (C.)	4.49 / 25.62	4.40 / 6.29	21.19 / 42.84	9.83 / 29.91
	Sim-C-by-C + Sim-C-by-R	3.66 / 23.78	3.14 / 13.84	15.95 / 42.55	7.68 / 29.19
	C. + Sim-C-by-C	8.23 / 25.15	2.52 / 13.21	20.90 / 46.67	12.10 / 31.39
	C. + Sim-C-by-R	8.78 / 23.50	8.49 / 10.06	21.84 / 40.47	13.05 / 28.25
	C. + Sim-C-by-C + Sim-C-by-R	10.84 / 27.44	4.40 / 6.92	19.45 / 44.59	13.21 _{+3.38} / 31.55 _{+1.64}

Table A4. 3D Object detection AP@0.3 on Carrada. The baseline is the RADDet model [60], which takes as input a 3D radar cube and predicts 2D boxes via a RA YOLO Head and 3D boxes via a RAD YOLO Head.

A.10. Comparison with RadSimReal and Standard Data Augmentation for Downstream Task

To further validate the quality of our simulation data, we compare the performance of Ctrl-RS with RadSimReal and standard data augmentation on 3D object detection. Ta-

ble A5 shows the comparison results on RADDet. Among the different radar simulation methods, although the model trained on the simulation data of RadSimReal gains a slight performance gain, it is still not as good as the model trained on the simulation data of Ctrl-RS. This fully illustrates the high qualitative requirements of training data for 3D object

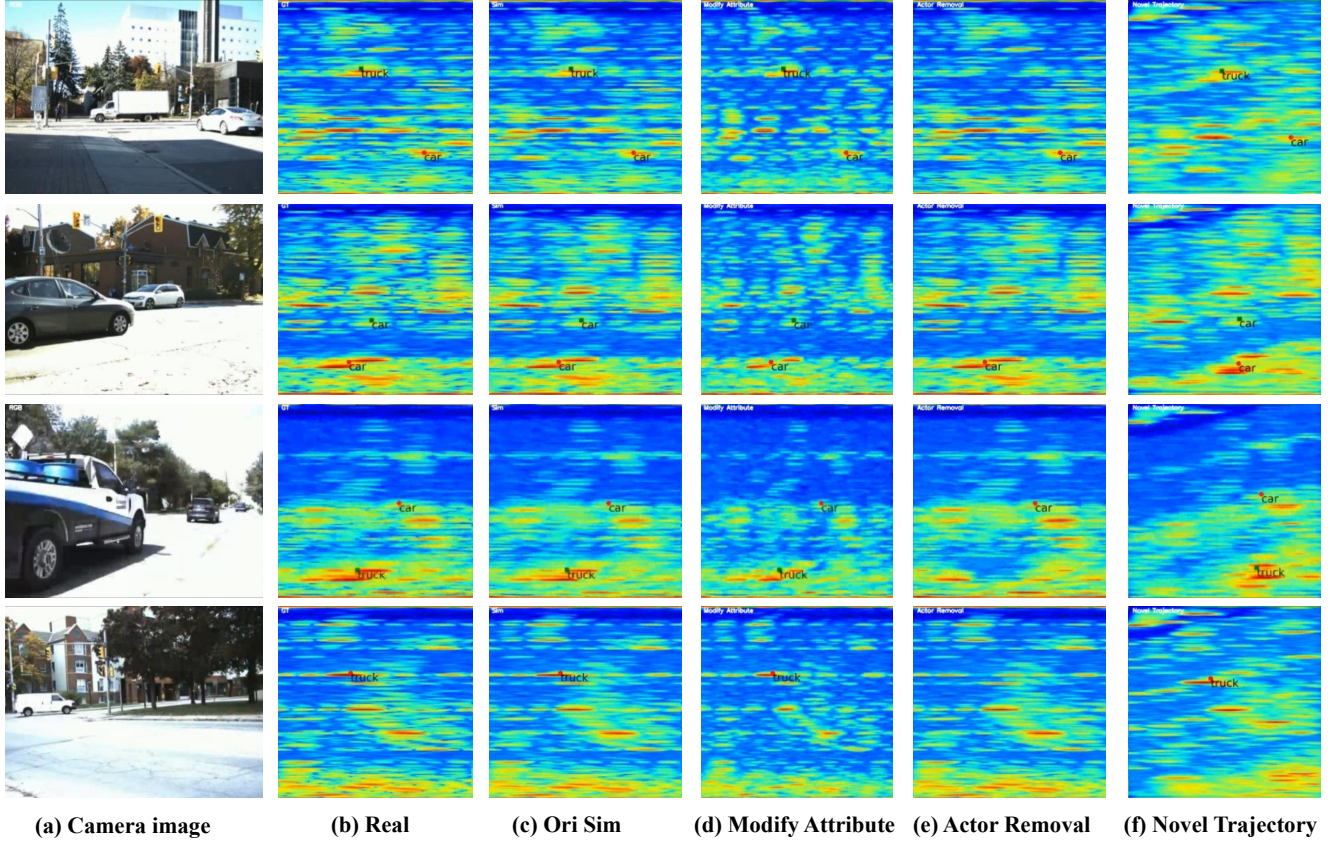


Figure A6. More results of radar simulation and scene editing.

Test set	Train set	AP@0.3					
		person	bicycle	car	bus	truck	all
RADDet 3D test set	RADDet (R.)	32.91	22.25	65.42	43.42	51.55	55.50
	R. + RadSimReal	33.76	22.25	65.20	38.16	54.99	55.83
	R. + Gaussian Noise	31.97	17.34	62.31	44.74	51.81	53.05
	R. + Random Rotation	30.42	16.81	65.62	28.19	47.55	54.13
	R.+ Sim-on-R-by-R + Sim-on-R-by-C	36.85	27.75	69.25	44.74	57.64	59.66

Table A5. Comparison of Ctrl-RS with RadSimReal and Standard Data Augmentation on 3D Object Detection. **Bold**: Best

detection, and effectively proves the physical rationality and accuracy of our Ctrl-RS’s simulation data.

Furthermore, the application of Gaussian noise and random rotation in the training leads to a performance loss for the detection model, as both destroy the physically relevant geometric representations in the radar data. We point out that, while the radar cube visually exhibits some randomness in the range and Doppler dimensions (which it is not), it is strictly smooth and morphologically identical in the azimuth dimension. Therefore, the diversity of the radar dataset cannot be enriched by simple data augmentation, which is quite different from the RGB image dataset.

A.11. Additional experimental results on unseen sensor and scenarios

We present the complete quantitative results of the 3D detection model on simulated data in Table A6. Additionally, we display the corresponding results from the RA perspective in Table A7. Specifically, we project the 3D bounding boxes of the detection model onto the range-azimuth dimension and then perform NMS processing to obtain 2D bounding boxes from the RA perspective.

Intuitively, significant performance variations exist across different datasets for 3D detection results of vari-

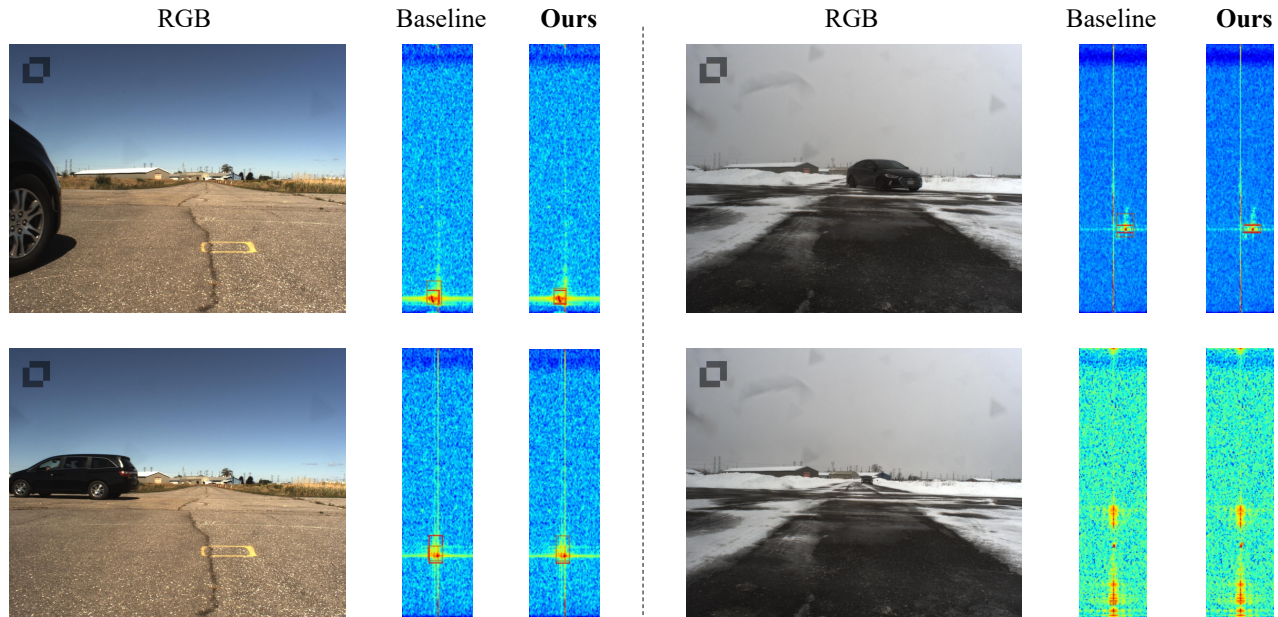


Figure A7. Visualization of 2D slices (range-Doppler) of radar cube from Carrada. The areas on both sides of the dashed line present RGB images from different scenes, along with the detection results on RD slices from both the baseline and our model (Ours). Specifically, the baseline model is trained only on real data, while our model is co-trained on our simulation data and real data. In each of the RD slice, red bounding boxes denote the ground truth, while brown bounding boxes represent the predictions.

ous categories. However, after grouping categories like car, bus, and truck into a single category, the quantitative results show a marked improvement. This indicates that errors in simulation data primarily stem from ambiguous object categorization. This is because the distinction between different object categories in radar simulations mainly lies in the distribution of reflection points, rather than reflection intensity (e.g., cars and buses exhibit similar reflection intensities).

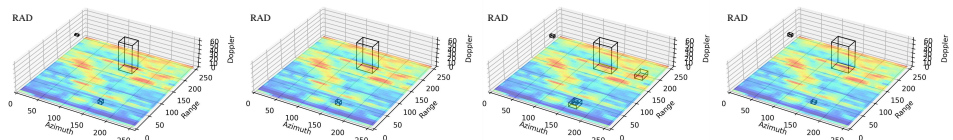
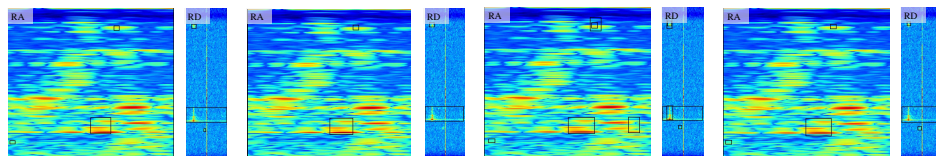
In contrast, from the RA perspective, the differences in results across categories and datasets diminish. This is because the distribution of reflection points in Doppler is directly related to the target’s radial velocity relative to the sensor, involving both relative position and relative velocity. However, lacking this information, we can only assign approximate Doppler values to the point cloud of each target. Consequently, when this dimension is ignored (considering only range and azimuth), quantitative metrics improve significantly. Similarly, treating cars, buses, and trucks as a single category further reduces differences across datasets.

Overall, we arrive at a crucial conclusion: even when starting from camera images or lidar point clouds, our method robustly simulates physically plausible radar data, though it remains influenced by the selection of reflected points—a key challenge faced by previous radar point cloud simulation approaches. However, from the perspective of practical autonomous driving applications, this influence

appears less critical, as we need only determine whether obstacles exist in the scene, not their specific category or velocity.



(a) Scene



(b) Ground Truth

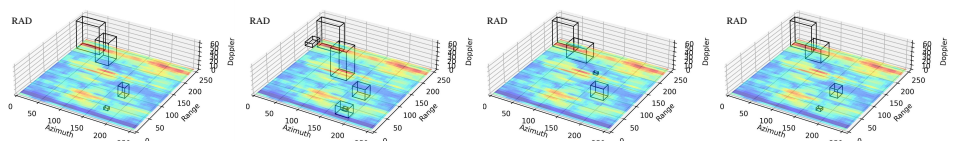
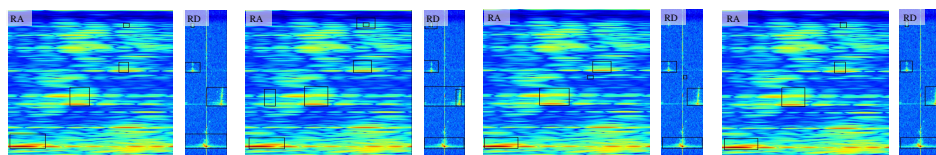
(c) Model trained on Real Data

(d) Model trained on Sim Data (Ours)

(e) Model trained on Real Data + Sim Data (Ours)



(a) Scene



(b) Ground Truth

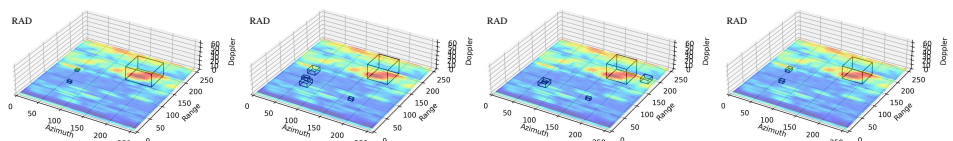
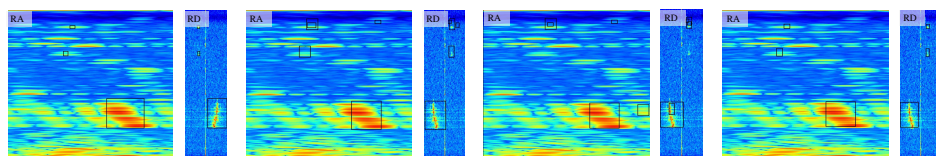
(c) Model trained on Real Data

(d) Model trained on Sim Data (Ours)

(e) Model trained on Real Data + Sim Data (Ours)



(a) Scene



(b) Ground Truth

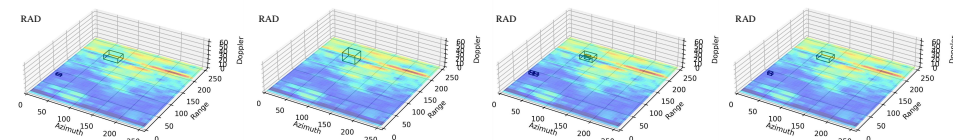
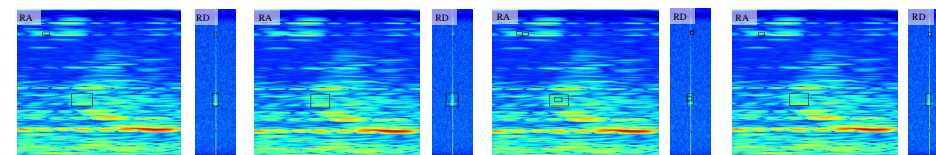
(c) Model trained on Real Data

(d) Model trained on Sim Data (Ours)

(e) Model trained on Real Data + Sim Data (Ours)



(a) Scene



(b) Ground Truth

(c) Model trained on Real Data

(d) Model trained on Sim Data (Ours)

(e) Model trained on Real Data + Sim Data (Ours)

Figure A8. Supplementary 3D detection results in the RADDet test set.

Test Set	Train Set	Person	Bicycle	Car	Bus	Truck	All
Sim-R _{test} -by-R (AP@0.3)	RADDet	25.13 25.13	13.69	69.98 56.75*	30.77	39.72	45.60 50.00*
	R. + Sim-R-by-R + Sim-R-by-C	32.03 32.03	17.24	67.84 69.44*	46.15	54.70	57.41 61.29*
Sim-N _{LIDAR} -by-R (AP@0.1)	RADDet	17.38 17.38	0	28.45 37.72*	1.18	21.21	22.11 33.59*
	R. + Sim-R-by-R + Sim-R-by-C	21.56 21.56	0	36.87 43.58*	2.36	21.90	26.03 38.32*
Sim-N _{Camera} -by-R (AP@0.1)	RADDet	6.05 6.05	1.59	23.91 39.29*	2.17	25.12	18.24 32.53*
	R. + Sim-R-by-R + Sim-R-by-C	7.13 7.13	2.38	29.20 41.70*	2.17	20.37	20.01 34.39*

Table A6. 3D Object detection AP of RAD boxes on simulated data on RADDet test set scenes and nuScenes [5] scenes. Sim-R_{test}-by-R: generated on RADDet test set scenes with the radar attribute from $\{\sigma, g, Rs, \lambda\}_R$. Sim-N-by-R: generated on nuScenes [5] v1-mini scenes with the radar attribute from $\{\sigma, g, Rs, \lambda\}_R$. *: treat the above categories as a single category.

Test Set	Train Set	Person	Bicycle	Car	Bus	Truck	All
Sim-R _{test} -by-R	RADDet	39.09 39.09	25.86	73.19 70.99*	38.46	45.70	58.16 63.24*
	R. + Sim-R-by-R + Sim-R-by-C	49.34 49.35	29.31	84.06 81.39*	30.77	56.10	68.63 73.90*
Sim-N _{LIDAR} -by-R	RADDet	33.10 33.10	15.52	52.93 58.51*	0.00	72.19	40.90 52.14*
	R. + Sim-R-by-R + Sim-R-by-C	42.26 42.26	8.19	67.90 67.61*	1.38	71.90	49.14 60.85*
Sim-N _{Camera} -by-R	RADDet	13.49 13.49	11.81	43.85 62.10*	1.09	93.15	34.87 48.96*
	R. + Sim-R-by-R + Sim-R-by-C	17.35 17.35	5.65	55.30 68.92*	2.17	79.44	37.87 54.36*

Table A7. 3D Object detection AP@0.3 of RA boxes on simulated data on RADDet test set scenes and nuScenes [5] scenes. Sim-R_{test}-by-R: generated on RADDet test set scenes with the radar attribute from $\{\sigma, g, Rs, \lambda\}_R$. Sim-N-by-R: generated on nuScenes [5] v1-mini scenes with the radar attribute from $\{\sigma, g, Rs, \lambda\}_R$. *: treat the above categories as a single category.