

Radio Frequency Ray Tracing with Neural Object Representation for Enhanced RF Modeling

Xingyu Chen
UC San Diego

xic063@ucsd.edu

Zihao Feng
UC San Diego

zif004@ucsd.edu

Kun Qian
University of Virginia

kunqian@virginia.edu

Xinyu Zhang
UC San Diego

xyzhang@ucsd.edu

Abstract

Radio frequency (RF) propagation modeling poses unique electromagnetic simulation challenges. While recent neural representations have shown success in visible spectrum rendering, the fundamentally different scales and physics of RF signals require novel modeling paradigms. In this paper, we introduce RFScape, a novel framework that bridges the gap between neural scene representation and RF propagation modeling. Our key insight is that complex RF-object interactions can be captured through object-centric neural representations while preserving the composability of traditional ray tracing. Unlike previous approaches that either rely on crude geometric approximations or require dense spatial sampling of entire scenes, RFScape learns per-object electromagnetic properties and enables flexible scene composition. Through extensive evaluation on real-world RF testbeds, we demonstrate that our approach achieves 13 dB improvement over conventional ray tracing and 5 dB over state-of-the-art neural baselines in modeling accuracy, while requiring only sparse training samples.

1. Introduction

Recent advances in neural rendering have transformed 3D scene understanding and reconstruction in computer vision [4, 13, 14, 17, 32, 34, 38–40, 42]. Methods such as Neural Radiance Fields [20] and 3D Gaussian Splatting [9] have demonstrated exceptional capabilities in modeling complex light transport and material interactions for photorealistic scene reconstruction. These advances in modeling optical frequencies suggest promising approaches for addressing challenges in radio frequency (RF) propagation modeling, as both domains share fundamental physical principles. RF signals, like visible light, follow Maxwell’s equations and exhibit similar wave phenomena, including reflection, refraction, and diffraction. However, the much longer wavelengths, smaller antenna aperture, and ability to penetrate common objects, result in more sophisticated interaction

with objects.

Accurate modeling of RF propagation remains a challenging problem in wireless systems with diverse use cases. For instance, RF simulation enables optimal deployment of access points [16] and intelligent reflectors [19, 26], by computing signal coverage across all possible deployment configurations. While industry-standard ray tracing algorithms [25, 28] approximate electromagnetic wave behavior through ray-object interactions, they often exhibit significant simulation-to-reality gaps due to simplified modeling of surface characteristics and material electromagnetic properties. Recent neural approaches in RF modeling, such as NeRF² [43], have shown potential in learning detailed RF-object interactions. However, these data-driven models require dense spatial sampling and complete model retraining when environmental conditions change, highlighting the need for more efficient and adaptable approaches.

In this paper, we introduce RFScape, a novel RF simulation framework that synergistically integrates the high-fidelity of neural representation and the flexibility and interpretability of ray tracing. The core novelty of RFScape lies in its approach of generating dedicated neural representations for individual objects within a scene and seamlessly importing these representations into a ray tracing pipeline. By implicitly modeling the minute structural details and complex material properties of objects using neural representations, RFScape captures the intricacies of RF signal propagation and interactions. Simultaneously, RFScape preserves the inherent flexibility of traditional ray tracing methods, enabling editing and dynamic updates of the simulated RF environment as objects are added, moved, or removed. To enable optimization of scene geometries in RFScape, differentiable ray-geometry intersections are needed for gradient-based optimization. We choose to represent geometries using Signed Distance Functions (SDFs), as they provide continuous and differentiable distance fields that are compatible with automatic differentiation, making them suitable for gradient-based optimization of object shapes and positions [2, 35, 36].

By decoupling object representations from ray propaga-

tion modeling, RFScape facilitates flexible editing of an RF simulation scene, such as modification of objects and configuration of radio hardware. This flexibility is highly desirable for the design and verification of wireless communication and sensing systems. Moreover, RFScape paves the way towards a modularized primitive library, analogous to the well-established practice of curating libraries of 3D mesh models in computer graphics. Such a library would enable large-scale, modularized RF simulation through pre-trained object representations, primed for plug-and-play integration into the design process of wireless systems.

We have implemented RFScape and performed extensive evaluation using various wireless testbeds, including WiFi radios (both sub-6 GHz and 60 GHz bands) and millimeter-wave (mmWave) radar. Our microbenchmark results demonstrate that RFScape generates RF signals that match the ground truth with high fidelity, achieving 13 dB improvement in RF-object characterization compared to conventional visual model based ray tracing. With merely 1.25 samples/sq ft of training data, RFScape outperforms the state-of-the-art (SOTA) NeRF² by around 5 dB in WiFi channel prediction, and even higher for mmWave. In addition, RFScape can adapt to changes in a scene with a small set of 3 to 5 additional data samples.

Our contributions can be summarized as follows:

- We propose RFScape, a novel RF simulation framework that reduces the sim-to-real gap of conventional ray tracing while maintaining flexibility.
- We design a novel neural representation that can capture detailed structure and material properties of objects while ensuring compatibility with ray tracing. We further present an efficient training strategy for learning these neural representations.
- We evaluate RFScape extensively across diverse real-world wireless environments. Our results demonstrate RFScape’s effectiveness and its potential to foster new applications of RF simulation.

2. Related Work

RF propagation and EM field simulation. RF signal propagation and environmental interaction simulation is essential for wireless communication and sensing system design. Full-wave simulation methods, including Finite Element Method (FEM) [8] and Finite-Difference Time-Domain (FDTD) [33], provide high-accuracy results but are computationally intensive, limiting their application to small-scale RF device design such as antenna development. For large-scale simulations, ray tracing methods [15, 25] offer computational efficiency by modeling electromagnetic waves using geometrical optics principles. Ray tracing simulation encompasses transmitters (Tx), receivers (Rx), and mesh-based object representations within a scene. The mesh representation comprises multiple polygonal elements

that approximate the object’s surface geometry. The simulation initiates with ray emission from each Tx across all directions, where angular resolution parameters define the ray density distribution. The simulator subsequently computes ray-mesh intersections and, at each intersection point, generates secondary rays based on the number of specified interactions. These secondary rays are propagated according to their interaction mechanisms (e.g., reflection, refraction, or diffraction). The simulator also computes the *directional radiance*, which quantifies the ray’s power distribution along specific directions post-interaction, determined by the material properties (e.g., permittivity and conductivity) of the intersected objects. Ray propagation terminates upon reaching either the Rx or a specified maximum interaction threshold. The simulator aggregates all received rays at the Rx through coherent combination, computing the relative phase differences based on their respective propagation path lengths.

The accuracy of ray tracing simulations depends critically on precise object geometry and material property models, which exceed the capabilities of conventional sensing systems. Notable challenges arise even for specular objects with simple geometric structures, as visual 3D reconstruction methods often fail to capture microscale surface features that, while visually insignificant, can substantially influence RF signal interactions.

Neural representations for EM simulation. Neural scene representations have established a significant position in computer vision and computer graphics, particularly following the introduction of Neural Radiance Fields (NeRF) [21]. NeRF implements neural networks to model two fundamental scene properties implicitly: the volumetric density at each spatial coordinate and the directional radiance (comprising color and intensity) emitted from that coordinate. Through learning these spatially and directionally dependent radiance functions from multiple input images, NeRF generates photorealistic novel viewpoints without explicit geometric or surface representations.

There have been efforts to adapt neural representation for RF applications [5, 18], with NeRF²[43] being a prominent example. This method models the entire scene as an implicit neural representation, preventing the clear delineation of object boundaries and their RF interaction characteristics. This limitation introduces constraints on system adaptability and increases susceptibility to environmental variations. While WiNeRT [23] advances this approach by incorporating neural network-based material reflection parameters into ray tracing simulations, its reliance on traditional 3D mesh representations remains insufficient for capturing detailed structural features of physical objects [31].

RFScape employs a learnable geometry representation with a neural material network to capture the complex and intrinsic RF properties of physical objects. Additionally,

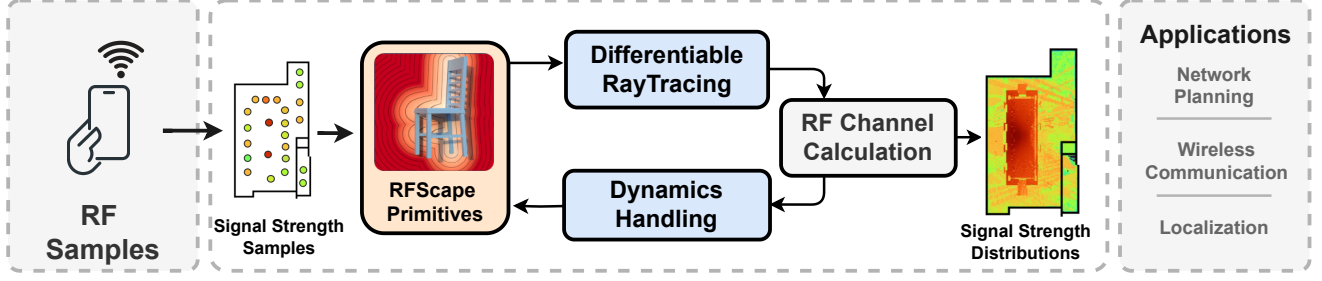


Figure 1. An overview of the RFScape framework.

RFScape separates empty space and medium to reduce optimization complexity and improve sampling efficiency.

3. Preliminaries

3.1. RF Ray Tracing

The proliferation of wireless communications and sensing systems has made accurate modeling of RF propagation increasingly critical. While full-wave electromagnetic (EM) solutions provide high accuracy, their computational complexity makes them impractical for large-scale scenarios. RF ray tracing offers an efficient alternative by applying geometric optics principles to electromagnetic wave propagation, enabling fast yet accurate predictions of radio coverage in complex environments. RF waves interact with the environment through reflection, refraction, diffraction, and scattering. These interactions are governed by Maxwell's equations but can be approximated using ray optics when the wavelength is much smaller than environmental features.

The workflow begins with environment modeling, where the physical space is discretized into polygonal surfaces with assigned electromagnetic properties. Each material i is characterized by its complex permittivity ε_i and conductivity σ_i :

$$\varepsilon_c = \varepsilon_i - j \frac{\sigma_i}{\omega \varepsilon_0}, \quad (1)$$

where ω represents the angular frequency and ε_0 is the free space permittivity. From each transmitter location $\mathbf{r}_t$, rays are launched at discrete angles (θ, ϕ) and traced according to Fermat's principle:

$$\frac{d}{ds} \left(n \frac{d\mathbf{r}}{ds} \right) = \nabla n, \quad (2)$$

where n denotes the refractive index of the medium, s represents the path length parameter, and $\mathbf{r}$ is the position vector.

3.1.1. RF Channel Calculation

The electric field $\mathbf{E}$ at receiver position $\mathbf{r}_r$ is computed by summing contributions from all paths:

$$\mathbf{E}(\mathbf{r}_r) = \sum_{p=1}^{N_p} A_p e^{-jk_0 s_p} \prod_{k=1}^{N_i} \mathbf{R}_k \cdot \mathbf{E}_0, \quad (3)$$

where the amplitude spreading factor A_p accounts for geometric attenuation, k_0 represents the free space wavenumber, s_p denotes the total path length, $\prod_{k=1}^{N_i} \mathbf{R}_k$ incorporates all reflection and transmission coefficients along the path, and $\mathbf{E}_0$ represents the reference field at unit distance. The reflection coefficient for perpendicular polarization follows:

$$R_{\perp} = \frac{\eta_2 \cos \theta_i - \eta_1 \cos \theta_t}{\eta_2 \cos \theta_i + \eta_1 \cos \theta_t}, \quad (4)$$

where η_i represents the wave impedance in medium i , and θ_i, θ_t are the incident and transmitted angles respectively.

3.2. Signed Distance Field (SDF)

Inspired by recent advances in computer graphics [35, 41], we propose to represent the geometric structure of an object using an SDF. Given an object Ω and its boundary $\partial\Omega$, an SDF $S_{\Omega}(p)$ defines the signed distance between any point p in the space and the object's surface $\partial\Omega$:

$$S_{\Omega}(p) = s(p, \Omega) \cdot d(p, \partial\Omega), \quad (5)$$

where $d(p, \partial\Omega) = \inf_{q \in \partial\Omega} d(p, q)$ denotes the minimal distance between the point p and any point q on the object's surface $\partial\Omega$. In addition, the sign $s(p, \Omega)$ indicates the *containing relation* between the point p and the object Ω , where a positive sign means that the point p is outside the object, *i.e.*, $p \notin \Omega$ while a negative sign means $p \in \Omega$. Intuitively, the SDF clearly defines the object's boundary, which are the solutions to the equation $S_{\Omega}(p) = 0$.

One fundamental property of SDF is the satisfaction of the Eikonal equation [30]:

$$\|\nabla S_{\Omega}(p)\| = 1. \quad (6)$$

It implies that the gradient of S_{Ω} at any point on the surface $\partial\Omega$ has a unit magnitude. Therefore, $\nabla S_{\Omega}(p)$ is exactly the normal vector n of the surface, which is necessary information for ray tracing to calculate new directions of RF rays

after interacting with an object. In practice, the normal vector $n(p)$ at any point p on the surface can be approximated by finite differences:

$$n(p) = \nabla S_{\Omega}(p) = \begin{bmatrix} \frac{S_{\Omega}(p+\epsilon_x) - S_{\Omega}(p-\epsilon_x)}{2\epsilon_x} \\ \frac{S_{\Omega}(p+\epsilon_y) - S_{\Omega}(p-\epsilon_y)}{2\epsilon_y} \\ \frac{S_{\Omega}(p+\epsilon_z) - S_{\Omega}(p-\epsilon_z)}{2\epsilon_z} \end{bmatrix}, \quad (7)$$

where ϵ_x , ϵ_y , and ϵ_z are small perturbations in the x , y , and z directions, respectively, and ϵ is the magnitude of these perturbations.

Following the above definition, every object occupying physical space has an associated SDF that implicitly encodes the object’s precise surface details. In addition, SDF defines a continuous signed distance field rather than discrete polygon surfaces, thus enabling gradient-based end-to-end optimization. However, SDF itself does not encode any material information, and lacks explicit definition of the object boundaries – both critical for RF ray tracing.

4. Method

RFScene is an RF simulation framework that integrates high-fidelity object models into a ray tracing engine, achieving both high accuracy and flexible editability for complex and even dynamic scenes. The framework consists of three primary components: 1) *RFScene primitives*, which model the geometric structures and material properties of objects for RF simulation; 2) *RF Ray Tracing*, which accurately simulates RF interactions; and 3) *Dynamics Handling*, which incorporates prior knowledge about scene changes to modify the RFScene scene. Additionally, RFScene’s optimization module enables end-to-end differentiable forward simulation and gradient-based backward optimization of the primitives’ parameters.

4.1. RFScene Primitives

A RFScene primitive represents an object in a scene to be simulated via ray tracing. Such a primitive must meet two requirements: 1) encoding of fine-grained geometric structure and material properties of the object, and 2) seamless interaction with RF rays. The first requirement can be fulfilled using trainable neural representations, such as NeRF [21, 43]. To further meet the second requirement, we need the neural representation to clearly define the boundary of the object, so that the geometric intersections between RF rays and objects can be localized. Unfortunately, as a scene-level model, NeRF obscures the clear delineation of individual objects. On the other hand, conventional mesh-based geometric representation suffers from inaccuracy (Sec. 2) and discontinuity, impeding its integration into an end-to-end optimization model.

4.1.1. Neural Representations of Geometric Structure and Material

To approximate the geometric structure of the object with fine-grained details, we propose to represent the SDF of the object using a *structure model*. As shown in Fig. 2, our structure model is a trainable neural network, which takes as input the position p of a marching ray and outputs a signed distance, subsequently used for sphere ray tracing. By learning the neural SDF representation, RFScene can eventually localize the interactions between rays and objects with infinitesimally small errors, potentially eliminating the sim-to-real discrepancies that plague conventional mesh representation.

However, only knowing the intersection of the ray and the object is insufficient for calculating the reflective and penetrative rays, which additionally depends on the local material properties of the object. Conventional RF ray tracing [15] usually calculates the Fresnel coefficients, assuming that the object has known and homogeneous material properties. Unfortunately, a practical object’s material properties are distributed unevenly across its surface and interior and are often not measurable. To overcome this challenge, we employ a second neural network to model the object’s material distribution and learn the directional attenuation coefficients caused by materials. As shown in Fig. 1, the inputs of this material model include any position p of the marching ray, the carrier frequency f_c of the RF signal, the local structure features f generated by the structure model, and the direction (θ, ϕ) of the resulting reflective or penetrative rays. The output is an attenuation coefficient α along that direction.

We emphasize that this model encapsulates all RF propagation effects (reflection, penetration, scattering, *etc.*) through the directional attenuation coefficient. These effects can all be represented by the attenuation along arbitrary directions after a marching ray interacts with a point on/within the object. In addition, unlike conventional ray tracing, where interactions only happen at the boundaries of objects, our material model accounts for the *interior* of the object. Specifically, when the position of a marching ray is inside the object, *i.e.*, $p \in \Omega$, the material model can output the attenuation coefficient for the propagation direction of the ray.

Following the success of the NeRF model [21], we implement both structure and material models as multi-layer perceptrons (MLPs). The interaction between a marching ray at p and the RFScene primitive of an object Ω can be formalized as follows:

$$M_{\Omega} : (p, \theta, \phi, f_c) \rightarrow (d, \alpha). \quad (8)$$

Both the signed distance d and the directional attenuation coefficient α are used to update the ray upon any interaction, including penetration and reflection along all possible

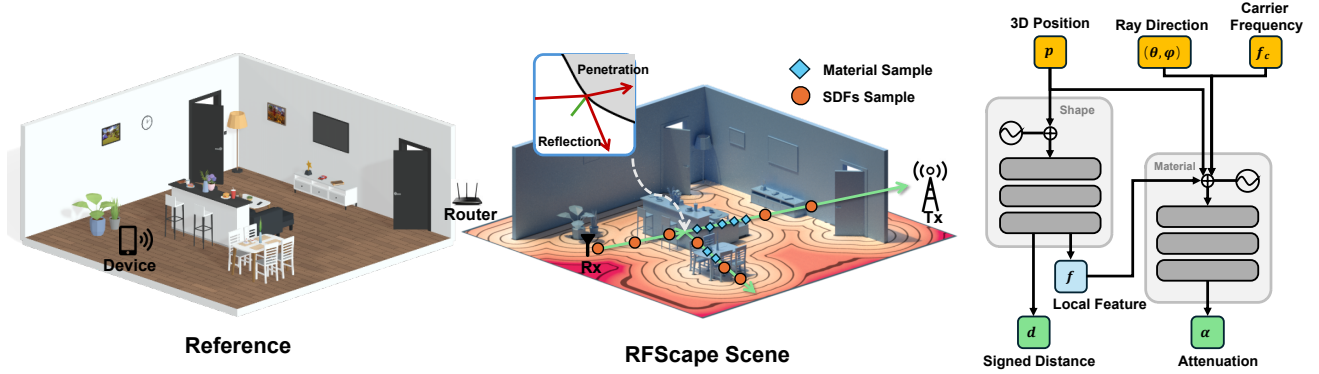


Figure 2. RF Ray Tracing with RFScape’s Neural Object Representations. While electromagnetic waves physically propagate from Tx (router) to Rx (device), the ray tracing computation traces paths backward from Rx to Tx for efficiency. Objects along these paths are modeled by our learned neural representations.

directions.

4.2. RF Ray Tracing with RFScape Primitives

Signal propagation along each ray path involves complex interactions with one or multiple objects. A vital step in ray tracing is to locate the interaction. Conventional ray tracing algorithms widely adopt the Möller–Trumbore algorithm [22] to compute the intersections between the rays and the polygons of a mesh-based object model. In contrast, *SDFs represent surfaces through continuous values without an explicit object boundary*. To overcome this challenge, RFScape locates the intersection point p by iteratively searching for the SDF zero-crossing along the ray direction. The iterative process is as follows:

$$T(p_0, \omega_0) = \sum_{i=0}^N S_{\Omega}(p_i), \quad p_{i+1} = p_i + \omega_0 \cdot S(p_i), \quad (9)$$

, where p_0 and ω_0 denote the initial position and propagation direction of the ray, and T , is the total distance traveled by the ray. The search terminates when $|S(p_i)| < \epsilon$, where $\epsilon = 0.01$, or when it reaches maximum iteration depths N . This is known as *sphere tracing* [7] for SDF-based object models.

Upon intersection, *i.e.*, $d = 0$, RFScape estimates the local normal vector n using Eq. (7) and determines the direction of the reflection ray, *i.e.*, $\omega_r = \omega_0 - 2(\omega_0 \cdot n)n$ according to Snell’s Law, and the direction of the penetrative ray, *i.e.*, $\omega_t = \omega_0$.

Given the directions of the rays after interactions, we can use the material model to obtain the attenuation coefficients for reflection and penetration, *i.e.*, a_r and a_t . To constrain the complexity of the RFScape ray tracing, we follow a Monte Carlo strategy and only select either of the reflective or penetrative ray for further propagation, with probabilities proportional to their directional attenuation coefficients.

Penetrating through an object, *i.e.*, $d < 0$, is similar to propagation in the free space, except that the attenuation coefficients are estimated using the material model and accumulated to obtain the total attenuation within the object, *i.e.*, $a_{\Omega} = \int_{t_{in}}^{t_{out}} M(p_0 + t \cdot \omega) dt$. When the interior ray reaches the boundary of the object again, the same Monte Carlo strategy is used to select either the internally reflective ray or the exiting ray.

Following the conventional ray tracing, RFScape emits rays from a Tx, calculating all interactions between the rays and the objects in the scene and eventually identifying the rays that reach the Rx to establish valid paths. Each valid path is characterized by its traversal distance τ and attenuation coefficient. We record all the intermediate segments of the rays and calculate the total distance τ and attention coefficient a along the path:

$$\tau = \sum_i t_i, \quad a = \prod_i a_i, \quad (10)$$

and the received signal can be calculated as:

$$S_{TX} = a \cdot S_{RX} \cdot e^{-j2\pi f_c \tau}. \quad (11)$$

4.3. RFScape Optimization

To capture the real structure and material properties of the objects and accurately represent the scene with RFScape primitives, we need to optimize their parameters Θ to minimize the sim-to-real discrepancy, *i.e.*,

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \sum_{i=0}^M \mathcal{L}(\hat{s}_i, s_i), \quad (12)$$

where $\hat{s}$ are RF measurements in the scene and M is the number of measurements. $\mathcal{L}$ is the loss that quantifies the discrepancy between real measurements $\hat{s}$ and RFScape’s

RF simulation output s . We expect that a few shots of RF measurements, *i.e.*, a small M , is sufficient since RFScape obviates the need to capture expansive empty regions and only focuses on objects in the scene. Besides, the physical laws of ray propagation pose extra constraints on the RFScape primitives of objects.

To promote the generation of continuous surfaces by the SDFs and ensure adherence to the eikonal constraints in Eq. (6), we further apply discrete Laplacian and eikonal regularization. This guarantees that the norm of the gradient remains unity, maintaining the physical plausibility of the neural SDF models.

4.4. Handling Scene Dynamics

Owing to the compatibility with conventional ray tracing, RFScape can support diverse radio configurations and dynamic scenes with little or even zero retraining efforts.

4.4.1. Tx/Rx Configurations

RFScape supports flexible configurations of radio antennas in simulation. Ideally, rays are shot uniformly in all directions with equal strength at Tx, while rays arriving at Rx are equally combined, assuming both Tx and Rx antennas are isotropic. For a practical antenna with known directivity (*i.e.*, antenna gain pattern), RFScape applies weights that are proportional to the antenna’s directive gains to rays along different directions. For a phased array that consists of multiple antenna elements, we can view the array as an equivalent directional antenna with switchable beam patterns. The radiance field of the antenna beam patterns can thus be simulated separately. By incorporating these antenna configurations into the ray tracing algorithm, we can directly reuse the RFScape primitives to simulate the performance of different radio configurations in a scene.

4.4.2. Object Dynamics

RFScape can handle dynamic scenes where objects can be added, removed, or moved. We assume that the RFScape primitives of newly added objects are available in a pre-trained RFScape library. Otherwise, they can be trained following the optimization procedure in Sec. 4.3, by freezing the already trained RFScape primitives of other objects in the scene. Unlike scene-wise neural representations [18, 43] that have to be completely retrained even upon partial environmental changes such as object locations, RFScape only needs to adapt to the partial changes, depending on how these changes are specified.

First, when visual sensors (*e.g.*, cameras or lidars) are available, RFScape can utilize them to extract the identities of moving objects and their poses (*i.e.*, translation and rotation vectors) with sufficient accuracy. Given the updated pose of an object, RFScape transforms the global position and directions of marching rays into the original coordinate system of this object before it moves, so that its RFScape

primitive can be used to model the interactions with the rays. This allows for updates of the scene based on the visual changes in the environment.

Second, when visual sensors are unavailable but RF measurements (*e.g.*, signal strength) can be sampled within the scene, we can *freeze* the RFScape primitives of the objects, and only optimize the poses of the foreground dynamic objects following Sec. 4.3. As inputs for the optimization, the RF measurements can be collected by devices already deployed within the scene.

We note that, similar to conventional ray tracing applications, it is also common that the scene dynamics are specified by users as part of an RF simulation process, *i.e.*, when specifying and iterating over possible base station locations to optimize coverage.

5. Experiment

5.1. Experimental Settings

Dataset We conduct real-world wireless measurements using a mobile robotic platform. We utilize a Turtlebot4 equipped with an integrated LiDAR sensor and Simultaneous Localization and Mapping (SLAM) system for precise navigation and positioning. The platform follows predefined trajectories with designated waypoints to ensure consistent data collection across the target space. For wireless signal acquisition, we deploy an ASUS RT-AC86U router operating in concurrent dual-band mode (2.4 GHz and 5 GHz) as the access point, paired with an iPhone 14 Pro client device leveraging the AirPort Utility application [1] for Received Signal Strength Indicator (RSSI) measurements. Additionally, our setup incorporates two 802.11ad-compliant MikroTik wAP 60G×3 routers [3] for 60 GHz WiGig measurements, running OpenWrt with Mikrotik Researcher Tools [6] to enable RSSI collection.

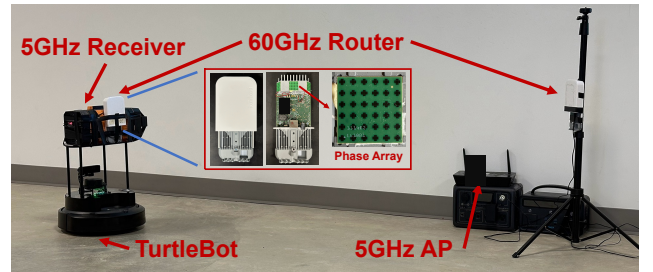


Figure 3. Room-level Wireless Channel Distribution Collection Setup.

Baselines We employ NeRF² [43] as the baseline, as it represents the state-of-the-art in neural channel prediction, outperforming other approaches such as Deep Convolutional Generative Adversarial Network (DCGAN)[27] and Variational Autoencoder (VAE)[11]. Our implementation follows the default configuration provided in the publicly

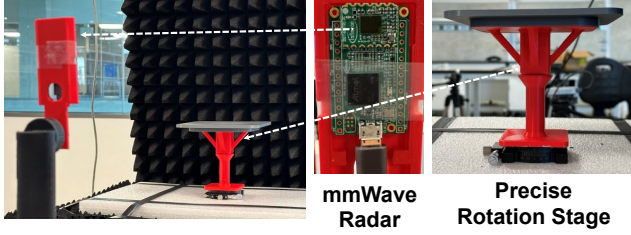


Figure 4. Object-level Wireless Channel Response Collection Setup.

available code archive. The reproduced results show a performance of 2.37 dB on BLE RSSI, 22.24 dB on WiFi SNR, and 0.82 SSIM for the RFID spectrum, which aligns with the original paper.

Model Implementation and Training

- **Structure model:** We employ an 8-layer Multilayer Perceptron (MLP) with a width of 64 and integration of skip connections. To address the spectral bias inherent in MLPs, the input 3D location x undergoes positional encoding with 6 frequencies. The actual implementation is $S_{\Omega} : x \rightarrow (d, f)$, where we output additional 128-dimensional local features f for the material model.
- **Material model:** The material model consists of an 8-layer MLP with a width of 256 and incorporates skip connections. The spatial inputs (x, n, θ) enhance the network’s sensitivity to spatial variations. This architecture is designed to encapsulate the intricate RF-material interactions, ensuring a faithful representation of material properties and electromagnetic wave behavior.

The training configuration includes a batch size of 4096 and utilizes the Adam optimizer [10]. The learning rate is initialized at 3×10^{-4} and gradually decays to 3×10^{-5} following an exponential schedule. Default values are maintained for other hyperparameters, such as $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. On an NVIDIA A6000 GPU, the network typically converges after approximately 300k iterations when training on a single scene.

5.2. Results

In this section, we conduct a microbenchmark experiment to evaluate RFScape’s capability in characterizing the interaction between RF signals and objects. We place individual objects on a precise rotation stage with RF absorbers around to isolate ambient multipath, as shown in Fig. 4. We position an FMCW radar 1m from the target objects to transmit mmWave signals and record the return signals. We rotate the objects at intervals of 5 or 10 degrees while recording the radar’s Range-FFT signals. We randomly select 50% of the collected data for training and reserve the remaining 50% for testing, ensuring uniform distribution across rotation angles. The evaluation includes three objects with increasing geometric complexity: a kettle, a teacup, and a

robot.

5.2.1. Object-level RF Characterization

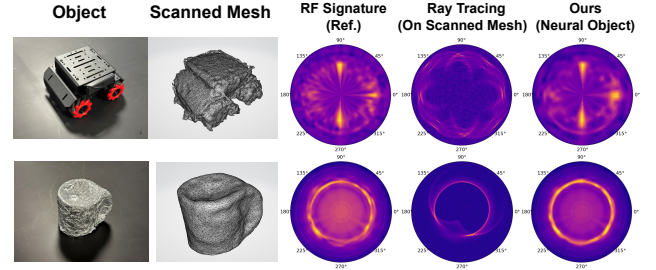


Figure 5. RF Characterization Prediction Results.

To establish a baseline, we employ Polycam [24], a highly optimized commercial 3D mesh scanner application [37], to capture mesh models of the objects. These models are then simulated using well-established ray tracing algorithms. The material parameters used in the simulations are obtained from relevant literature [12, 29].

The results are shown in Fig. 5. The mesh models obtained from visual scans fail to capture accurate microgeometry. They also assume objects have homogeneous materials throughout. Consequently, directly applying ray tracing on visually scanned meshes yields significant errors, with a large median error of 15.7 dB. In contrast, RFScape leverages neural object representations to model the RF signatures of objects using only around one sample/sq ft of RF observations, reducing the median error to just 2.9 dB while still preserving the scene editability. This makes RFScape a much more accurate option for RF object modeling compared to the mesh-based approaches commonly used in traditional ray tracing. It is worth noting that NeRF² did not demonstrate learning from FMCW radar signals, and the objects fit by NeRF² cannot be edited or re-simulated, hence its exclusion from this experiment.

5.2.2. Room-Level RF Channel Prediction Evaluation

To evaluate the performance and scalability of RFScape in predicting the RF channel distribution, we compare it with the state-of-the-art NeRF²[43] framework. The experiments are conducted with 5-fold cross-validation using the dataset provided by NeRF², which includes 2.4 GHz WiFi CSI, BLE RSSI, and RFID spectrum data [43], as well as our self-collected 60GHz WiGig CSI and additional 5GHz WiFi data (Sec. 5.1). NeRF² requires approximately 200 channel measurements per square foot as training data, which presents significant limitations in practical applications. To evaluate performance under realistic conditions, we evaluate both RFScape and NeRF² across multiple training data densities: very small (0.625 samples/sq ft), small (1.25 samples/sq ft), medium (2.5 samples/sq ft), large (5 samples/sq ft), and very large (10 samples/sq ft).

Fig. 6, 7, and 8 present the comparison results between RFScape and NeRF² under different wireless protocols.

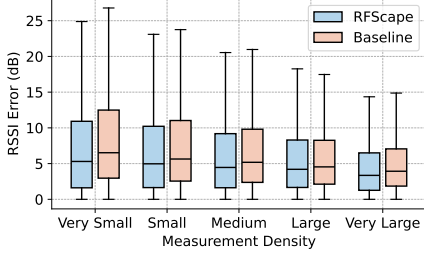


Figure 6. BLE RSSI (Lower values indicate better performance)

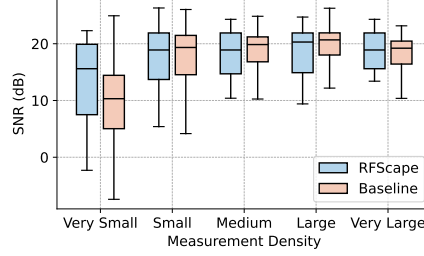


Figure 7. 2.4 GHz WIFI CSI (Higher values indicate better performance)

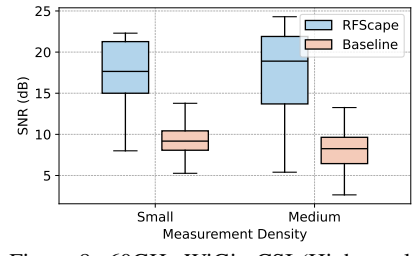


Figure 8. 60GHz WiGig CSI (Higher values indicate better performance)

For the small-sized WiFi CSI training dataset, RFscape achieves a median SNR of 15.6 dB, outperforming NeRF² by 5.3 dB. Similarly, on the small-sized RFID spectrum datasets, RFscape is comparable to NeRF², with SSIMs of 0.4498 and 0.5061, respectively. On the small-sized BLE RSSI datasets, RFscape reduces the RSSI error to 5.3 dB, which is 1.2 dB better than the baseline.

These results suggest that *RFscape is more efficient in learning and generalizing from limited data samples*. This is due to better physical constraints and clear object boundaries that eliminate the need to capture expansive empty regions.

The advantage of RFscape becomes even more pronounced at higher frequency bands, as demonstrated in Fig. 8. The narrower beams, shorter wavelengths (hence less diffraction), and higher path loss of the mmWave signals together result in sparser signal propagation compared to lower frequency bands. Accurately predicting the mmWave channel requires finer-grained physical simulation, which is incorporated in RFscape. In contrast, the baseline NeRF² struggles to predict mmWave band CSI, resulting in a low SNR of 9.08 dB. RFscape, on the other hand, maintains a 16.2 dB SNR in CSI channel prediction for the mmWave band. These results indicate that *RFscape yields superior performance on high-frequency channels which are more sensitive to ray-object interactions*.

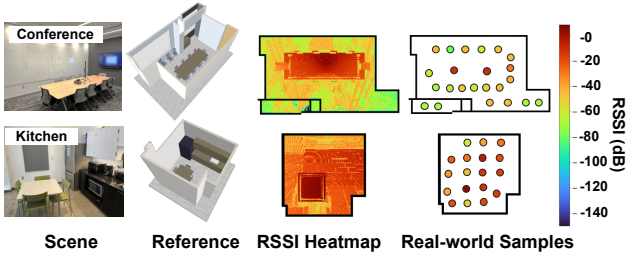


Figure 9. Channel Prediction Results.

5.2.3. Dynamics Adaptation Evaluation

In this section, we evaluate the adaptability of RFscape to environmental alterations. The original scenario consists of a long wooden table with chairs positioned along its length,

and additional plastic chairs are located in the corner of the room. We consider a scenario where, after a conference discussion, the scene is modified by introducing a blackboard and repositioning several chairs.

To facilitate RFscape’s adaptation to the modified environment, we evaluate the two approaches proposed in Sec. 4.4.2. The first method involves utilizing assertions based on visual information. We assume that information regarding the location, orientation, and type of objects that are added, changed, or removed is obtained from a camera in the room. The second approach involves *recollecting a small set of 3 to 5 data points* from the modified scenario.

Subsequently, we re-scan the environment using the setup shown in Fig. 3 as ground truth. Both methods successfully update RFscape’s scene representation and accurately predict the updated scene channels, with median RSSI errors of 2.9 and 3.2 dB, respectively. These minimal discrepancies between the adapted RFscape and the ground truth demonstrate RFscape’s adaptability to scene dynamics with minimal assisted information.

6. Conclusion

We have presented RFscape, a novel framework that seamlessly integrates high-fidelity neural object representations with ray tracing for accurate and flexible RF simulations. RFscape is inspired by state-of-the-art SDF-based object modeling techniques in computer graphics, but overcomes their incompatibility with ray tracing and lack of models for RF-material interaction. The modular nature of RFscape enables reconfiguration of the RF simulation scenes with little or zero site-specific retraining. As an advancement to the physics-based ray tracing simulators widely used in the wireless industry, we believe RFscape can serve as a powerful computer-aided design tool for diverse wireless applications, such as network planning and inverse simulation for 3D reconstruction. We leave the exploration of such applications for future work.

Acknowledgements

This research was partly supported by Samsung Research America and by NSF under Grants CNS-2128588, CNS-2312715, and CNS-2403124.

References

- [1] Apple. Airport utility, 2011. 6
- [2] Sai Praveen Bangaru, Michael Gharbi, Fujun Luan, Tzu-Mao Li, Kalyan Sunkavalli, Milos Hasan, Sai Bi, Zexiang Xu, Gilbert Bernstein, and Fredo Durand. Differentiable rendering of neural sdfs through reparameterization. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 1
- [3] Alejandro Blanco, Pablo Jiménez Mateo, Francesco Gringoli, and Joerg Widmer. Augmenting mmwave localization accuracy through sub-6 ghz on off-the-shelf devices. In *Proceedings of the ACM MobiSys*, 2022. 6
- [4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 1
- [5] Xingyu Chen, Zihao Feng, Ke Sun, Kun Qian, and Xinyu Zhang. Rfcanvas: Modeling rf channel by fusing visual priors and few-shot rf measurements. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, pages 464–477, 2024. 2
- [6] IMDEA Networks Wireless Networking Group. Researcher version of openwrt for wap 60g/lhgg-60ad/wap 60gx3 ap, 2024. 6
- [7] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996. 5
- [8] Jian-Ming Jin. *The finite element method in electromagnetics*. John Wiley & Sons, 2015. 2
- [9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1
- [10] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7
- [11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6
- [12] B. Langen, G. Lober, and W. Herzig. Reflection and transmission behaviour of building materials at 60 ghz. In *5th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Wireless Networks - Catching the Mobile Future.*, pages 505–509 vol.2, 1994. 7
- [13] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4273–4284, 2023. 1
- [14] David B Lindell, Julien NP Martel, and Gordon Wetzstein. Autoint: Automatic integration for fast neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14556–14565, 2021. 1
- [15] Hao Ling, R-C Chou, and S-W Lee. Shooting and bouncing rays: Calculating the rcs of an arbitrarily shaped cavity. *IEEE Transactions on Antennas and Propagation*, 37(2):194–205, 1989. 2, 4
- [16] An Liu and Vincent KN Lau. Optimization of multi-uav-aided wireless networking over a ray-tracing channel model. *IEEE Transactions on Wireless Communications*, 18(9):4518–4530, 2019. 1
- [17] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 1
- [18] Haofan Lu, Christopher Vatheuer, Baharan Mirzasoleiman, and Omid Abari. A deep learning framework for wireless radiation field reconstruction and channel prediction, 2024. 2, 6
- [19] Ruichun Ma, Shicheng Zheng, Hao Pan, Lili Qiu, Xingyu Chen, Liangyu Liu, Yihong Liu, Wenjun Hu, and Ju Ren. Automs: Automated service of mmwave coverage optimization using low-cost metasurfaces. In *Proceedings of the ACM MobiCom*, 2024. 1
- [20] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1
- [21] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2, 4
- [22] Tomas Möller and Ben Trumbore. Fast, minimum storage ray/triangle intersection. In *ACM SIGGRAPH 2005 Courses*, pages 7–es. 2005. 5
- [23] Tribhuvanesh Orekondy, Pratik Kumar, Shreya Kadambi, Hao Ye, Joseph Soriaga, and Arash Behboodi. Winert: Towards neural ray tracing for wireless channel modelling and differentiable simulations. In *Proceedings of the ICLR*, 2022. 2
- [24] polycam. Polycam - LiDAR & 3D Scanner for iPhone & Android. <https://poly.cam/>, 2023. 7
- [25] Telecom Infra Project. Open afc. <https://telecominfraproject.com/open-afc/>, 2023. 1, 2
- [26] Kun Qian, Lulu Yao, Xinyu Zhang, and Tse Nga Ng. Millimirror: 3d printed reflecting surface for millimeter-wave coverage expansion. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 15–28, 2022. 1
- [27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 6
- [28] Remcom. Wireless insite 3d wireless propagation software. <https://www.remcom.com/wireless-insite-propagation-software>, 2024. 1
- [29] K. Sato, T. Manabe, T. Ihara, H. Saito, S. Ito, T. Tanaka, K. Sugai, N. Ohmi, Y. Murakami, M. Shibayama, Y. Konishi, and T. Kimura. Measurements of reflection and transmission characteristics of interior structures of office building in the 60-ghz band. *IEEE Transactions on Antennas and Propagation*, 45(12):1783–1792, 1997. 7

- [30] James A Sethian. A fast marching level set method for monotonically advancing fronts. *proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. [3](#)
- [31] Dario Seyb, Eugene d’Eon, Benedikt Bitterli, and Wojciech Jarosz. From microfacets to participating media: A unified theory of light transport with stochastic geometry. *ACM Transactions on Graphics (TOG)*, 43(4):1–17, 2024. [2](#)
- [32] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8269–8279, 2022. [1](#)
- [33] Allen Taflove, Susan C Hagness, and Melinda Picket-May. Computational electromagnetics: the finite-difference time-domain method. *The Electrical Engineering Handbook*, 3 (629-670):15, 2005. [2](#)
- [34] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4):1–12, 2019. [1](#)
- [35] Delio Vicini, Sébastien Speierer, and Wenzel Jakob. Differentiable signed distance function rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)*, 41(4):125:1–125:18, 2022. [1](#), [3](#)
- [36] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. [1](#)
- [37] Zeyu Wang, Cuong Nguyen, Paul Asente, and Julie Dorsey. Pointshopar: Supporting environmental design prototyping using point cloud in augmented reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2023. [7](#)
- [38] Fanbo Xiang, Zexiang Xu, Milos Hasan, Yannick Hold-Geoffroy, Kalyan Sunkavalli, and Hao Su. Neutex: Neural texture mapping for volumetric neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7119–7128, 2021. [1](#)
- [39] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8254–8263, 2023.
- [40] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5752–5761, 2021. [1](#)
- [41] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. [3](#)
- [42] Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. NeRFusion: Fusing Radiance Fields for Large-Scale Scene Reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [1](#)
- [43] Xiaopeng Zhao, Zhenlin An, Qingrui Pan, and Lei Yang. Nerf2: Neural radio-frequency radiance fields. In *Proceedings of the ACM MobiCom*, 2023. [1](#), [2](#), [4](#), [6](#), [7](#)