# RFWEAVE: LEARNING WIRELESS PROPAGATION ACROSS SCENES AND TRANSCEIVER CONFIGURATIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Wireless propagation modeling is foundational to the evaluation and design of modern communication systems. To achieve accurate wireless propagation prediction, recent learning-based approaches leverage scene geometry and materials, demonstrating promising generalization across transceiver locations and unseen scenes. However, these approaches remain largely confined to a fixed configuration of array size, receiver orientation, and transmit beam. In this work, we introduce RFWeave, a learned wireless propagation model designed to generalize across scenes, transceiver locations, and transceiver configurations. To accommodate varying array structures, RFWeave incorporates variable-length transceiver tokens that represent antenna elements individually. Furthermore, to capture the geometry between transmit and receive antenna elements, RFWeave aggregates their pairwise spatial relations into propagation tokens. A configuration-conditioned decoder predicts the spatial spectrum and received power. RFWeave is trained on a large-scale simulated dataset spanning 10,000 indoor scenes and more than 5 million channel records under diverse transceiver configurations. Extensive evaluations show that RFWeave achieves state-of-the-art performance across all evaluated generalization settings, improving spatial spectrum structural similarity (SSIM) by up to 15.7% and reducing received power mean absolute error (MAE) by up to 35.4% over the strongest baselines on unseen scenes. Additional experiments on measured DICHASUS data and in beam management further demonstrate RFWeave's broader applicability to real-world channels and downstream wireless tasks.

## 1 INTRODUCTION

For modern communication, accurate wireless propagation modeling is foundational to link-level evaluation and system design. It characterizes how radio-frequency (RF) signals propagate through physical environments and are observed at receivers. High-fidelity propagation prediction requires accounting for scene geometry and materials, transceiver locations, and transceiver configurations. While current learned propagation models can generalize across transceiver locations (Zhao et al., 2023; Wen et al., 2025), they typically require scene-specific optimization and assume a fixed transceiver configuration, limiting direct reuse across changing environments.

In pursuit of generalizable wireless propagation prediction, prior work has largely followed two main directions, but each remains limited in scope. Cross-scene propagation methods learn to predict wireless propagation from 3D scene geometry and material properties (Hehn et al., 2025; Yang et al., 2026; Wen et al., 2026). However, they typically assume a common transceiver configuration, with array size, receiver orientation, and transmit beam held fixed. In parallel, transceiver configuration has mainly been studied in downstream tasks such as beam prediction, selection, and beamforming (Xu et al., 2020; Rezaie et al., 2022a;c; Kilinc & Heath, 2026; Bian et al., 2026). These methods incorporate antenna orientation, array structure, beam information, or antenna heterogeneity to improve communication decisions. Yet these lines of work have largely treated scene and transceiver variation as separate problems. In practice, these factors often vary together: mobile transceivers continuously change location and orientation, transmit beams are dynamically reconfigured, and deployment environments vary across scenarios. This motivates a unified propagation model that generalizes jointly across scenes, transceiver locations, and transceiver configurations.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

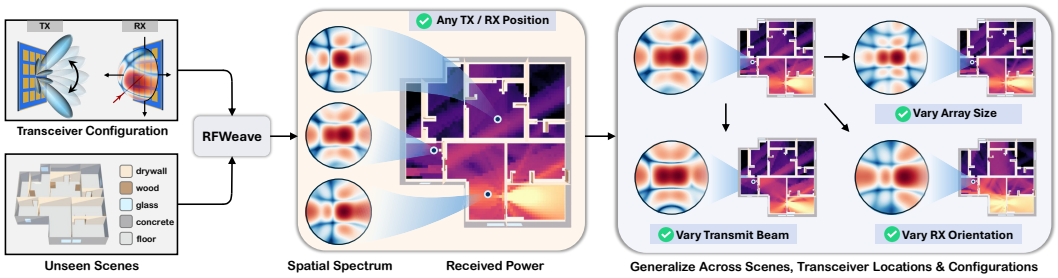


Figure 1: **RFWeave.** Given a material-annotated scene, queried TX/RX locations, and a transceiver configuration, RFWeave predicts the AoA spatial spectrum and received power using a single model. The same model supports arbitrary TX/RX location queries and generalizes across array sizes, receiver orientations, transmit beams, and their combinations on unseen scenes. The received power heatmap is obtained by querying RFWeave at multiple receiver locations.

Beyond scene and location generalization, transceiver configuration changes introduce a distinct representation challenge. Array size changes both the number and spatial arrangement of antenna elements. Receiver orientation changes the geometric relationship between the array and its surroundings, whereas the transmit beam changes the directional excitation of the scene. These configuration factors may also vary jointly at test time. Consequently, changes in transceiver configuration affect not only the values of model inputs, but also the structure and geometry that the model must represent.

In this work, we introduce RFWeave, a learned wireless propagation model designed for joint generalization across scenes, transceiver locations, and transceiver configurations (Fig. 1). Building on recent work in generative modeling of radio maps and wireless channel distributions (Wang et al., 2025; Böck et al., 2025), RFWeave models the receive-side response through spatial covariance and received power, rather than reproducing a full instantaneous complex channel. The covariance captures spatial structure across receive antenna elements, while received power characterizes the overall signal strength. The angle-of-arrival (AoA) spatial spectrum can then be computed from the covariance, providing a directional characterization of the receive-side propagation response.

To accommodate varying array structures, RFWeave represents transceivers with variable-length tokens that preserve antenna element structure across array sizes. The material-annotated scene point cloud is compressed into scene tokens that summarize local geometry and material properties. RFWeave further encodes the pairwise geometry between transmitter (TX) and receiver (RX) elements and aggregates these interactions into element-indexed propagation tokens. A scene-conditioned backbone then refines the propagation tokens using the scene and transceiver representations. Because receiver orientation and transmit beam are introduced only at the prediction stage, the backbone representation can be reused across multiple configuration queries.

We construct a comprehensive wireless propagation dataset by applying far-field synthesis to multipath channel records from WiPTR (Hehn et al., 2025) and Wi3R (Orekondy et al., 2023), yielding spatial spectrum and received power targets. The dataset contains 10,000 scenes and over 5 million channel records. Training covers 6 array sizes from $3 \times 3$ to $8 \times 8$, 8 receiver orientations at $45°$ intervals, and 16 transmit beams, while evaluation includes a wide range of unseen configurations. These data span variations in scenes, transceiver locations, and transceiver configurations, enabling RFWeave to learn a generalizable model of wireless propagation. Beyond simulation, we use measured DICHASUS channels (Euchner et al., 2021) to study adaptation under limited measured data. We also evaluate RFWeave on a beam management task in a simulated factory (Bian et al., 2026).

Our experiments demonstrate that RFWeave generalizes across scenes, transceiver locations, and transceiver configurations. On unseen scenes, RFWeave improves spatial spectrum SSIM by up to 15.7% and reduces received power MAE by up to 35.4% over the strongest baselines. These gains persist under unseen array sizes, receiver orientations, and transmit beams, as well as joint configuration variation. We further validate RFWeave on measured DICHASUS channels, where it adapts effectively with as few as 100 measured samples. In downstream beam management, RFWeave enables beam decisions that achieve 34.5% higher rates than location-based beam steering. Overall, our results highlight the generalizability and downstream utility of RFWeave.

## 2 Related Work

**Learned wireless propagation modeling.** Learned propagation modeling has been studied under both scene-specific and cross-scene settings. In scene-specific modeling, neural-field and Gaussian-based approaches reconstruct wireless radiation fields from scene geometry, sparse measurements, or both, enabling propagation prediction at unobserved locations (Zhao et al., 2023; Lu et al., 2024; Wen et al., 2025; Umer et al., 2026). Other approaches learn electromagnetic properties of the environment and integrate them with differentiable ray tracing, but still require detailed 3D scene models and incur the computational cost of ray tracing (Chen et al., 2025; Ai et al., 2026). These formulations are typically built or adapted for a target environment. Cross-scene methods instead learn transferable propagation models from multiple environments and predict wireless characteristics in unseen scenes (Hehn et al., 2025; Yang et al., 2026; Wen et al., 2026). However, these methods typically assume a common transceiver configuration across scenes, so the effects of antenna configuration are learned implicitly under a fixed setup. We consider the more general setting where scene, transceiver locations and configurations vary jointly.

**Transceiver-aware wireless learning.** Transceiver variation has mainly been studied in downstream communication tasks. Prior approaches incorporate receiver orientation or multi-panel antenna structure for beam prediction and selection (Rezaie et al., 2022a;c). More recent work studies beam selection under heterogeneous antenna configurations, including variations in array size, orientation, codebooks, element patterns, and polarization (Rezaie et al., 2022b; Kilinc & Heath, 2026). Environment-aware approaches also use 3D scene representations to support beam prediction and beamforming (Xu et al., 2020; Bian et al., 2026). These methods account for transceiver variation, but primarily target downstream communication tasks rather than wireless propagation modeling.

## 3 Method

### 3.1 Problem Formulation

We consider a learned forward model of wireless propagation in which the receive-side response depends jointly on the scene $\mathcal{X}$, transceiver locations $\mathcal{Q}$, and transceiver configuration $\mathcal{C}$. The TX array contains $N_{\mathrm{t}}$ antenna elements, while the RX array contains $N_{\mathrm{r}}$ antenna elements.

We characterize the receive-side response through its spatial covariance. Under standard wide-sense stationary channel models, second-order spatial statistics capture correlations across array elements and relate closely to the angular distribution of received energy (Yang et al., 2023). Spatial covariance also avoids requiring a consistent absolute phase reference for instantaneous channel realizations. We normalize the covariance to separate its spatial structure from the overall signal scale, and model the latter through received power. With learnable parameters $\theta$, RFWeave learns the mapping

$$f_\theta(\mathcal{X}, \mathcal{Q}, \mathcal{C}) \mapsto \left(\widehat{\mathbf{R}}, \widehat{P}\right), \tag{1}$$

Here, $\widehat{\mathbf{R}} \in \mathbb{C}^{N_{\mathrm{r}} \times N_{\mathrm{r}}}$ and $\widehat{P}$ denote the normalized receive-side spatial covariance and received power, respectively. The spatial spectrum is obtained from $\widehat{\mathbf{R}}$ through steering vector evaluation, as described in Sec. 3.4. Our goal is a single model that generalizes across scenes, transceiver locations, and transceiver configurations, including joint variations across these factors.

### 3.2 Scene and Transceiver Tokenization

Geometry relative to the scene is expressed in a common global frame, whereas array-specific quantities such as element positions and beam directions are defined in local array frames. We use a global scene frame $G$, and let $T$ and $R$ denote local TX and RX frames anchored at the corresponding array centers. Accordingly, ${}^{G}\mathbf{p}$, ${}^{T}\mathbf{p}$, and ${}^{R}\mathbf{p}$ denote positions expressed in $G$, $T$, and $R$.

**Scene tokenizer.** We represent the scene as a material-annotated point cloud, with each point described by its position and surface normal in the global frame and by material properties including relative permittivity and conductivity. We employ a PointNet++-style set abstraction hierarchy (Qi et al., 2017; Qian et al., 2022), a widely used approach for hierarchical point cloud representation

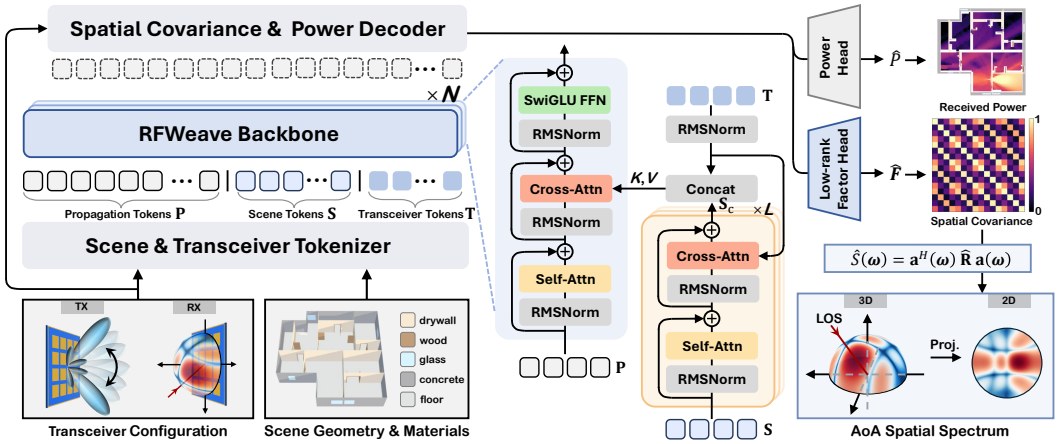


Figure 2: **Overview of RFWeave.** RFWeave organizes scene and transceiver information into structured tokens and refines element-indexed propagation tokens with a scene-conditioned backbone. The prediction stage outputs received power and a complex covariance factor, from which the spatial covariance and AoA spatial spectrum are obtained.

that has also been adopted in wireless learning tasks (Wen et al., 2026). Three abstraction stages progressively downsample the point set by factors of 4, 4, and 2, respectively, using farthest point sampling. At each stage, neighboring points are grouped using geometry, surface normals, and material properties, and their features are aggregated through attention pooling. The resulting $N_{\mathrm{s}}$ local features are projected to the common token dimension $d$, yielding $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_{N_{\mathrm{s}}}] \in \mathbb{R}^{N_{\mathrm{s}} \times d}$. We refer to them as ***material-aware scene tokens***.

**Transceiver tokenizer.** To accommodate different array sizes and element arrangements, we represent each antenna element as an individual token using its local position normalized by the carrier wavelength $\lambda$, with ${}^{T}\mathbf{p}_i/\lambda$ for TX element $i$ and ${}^{R}\mathbf{p}_j/\lambda$ for RX element $j$. Following the TX–RX link tokenization in Wi-GATr (Hehn et al., 2025), we additionally introduce tokens for the TX and RX array centers and their center-to-center displacement. Together, these tokens capture transceiver geometry at the element, array, and link levels and form a variable-length sequence $\mathbf{T} \in \mathbb{R}^{(N_{\mathrm{t}}+N_{\mathrm{r}}+3)\times d}$, naturally accommodating different array sizes. We refer to them as ***transceiver tokens***. Additional details on token projection and type encoding are provided in Appendix A.2.

### 3.3 Wireless Propagation Modeling

**Pairwise geometry encoding.** The transceiver tokens in $\mathbf{T}$ capture geometry at the element, array, and link levels, but do not represent the pairwise geometry between individual TX and RX elements. For each TX element $i$ and RX element $j$, we compute their global positions ${}^{G}\mathbf{p}^{\mathrm{t}}_i$ and ${}^{G}\mathbf{p}^{\mathrm{r}}_j$ under fixed reference orientations and the displacement ${}^{G}\mathbf{p}^{\mathrm{r}}_j - {}^{G}\mathbf{p}^{\mathrm{t}}_i$ between them. To account for carrier-scale geometry, we normalize the local element positions, relative displacement, and its magnitude by the carrier wavelength $\lambda$, and feed them together with their global positions and the corresponding unit direction into a learned projection, producing a pairwise feature $\mathbf{e}_{ij} \in \mathbb{R}^d$. Collecting all pairwise features gives a tensor $\mathbf{E} = [\mathbf{e}_{ij}]_{i=1,j=1}^{N_{\mathrm{t}},N_{\mathrm{r}}}$, whose size scales with $N_{\mathrm{t}}N_{\mathrm{r}}$.

However, keeping all $N_{\mathrm{t}}N_{\mathrm{r}}$ pairwise features throughout the backbone incurs substantial memory and computational costs for large arrays. We aggregate $\mathbf{E}$ along both the TX and RX dimensions. For each TX element $i$, a TX-indexed query attends to $\mathbf{E}_{i,:}$ as keys and values to summarize its pairwise relations with all RX elements, while the RX side is constructed symmetrically from the columns $\mathbf{E}_{:,j}$. These bidirectional summaries form $N_{\mathrm{t}} + N_{\mathrm{r}}$ element-indexed ***propagation tokens*** $\mathbf{P} \in \mathbb{R}^{(N_{\mathrm{t}}+N_{\mathrm{r}})\times d}$. Further aggregation details are provided in Appendix A.3.

**Scene conditioning.** As shown in Fig. 2, to incorporate scene context, we first refine the scene tokens $\mathbf{S}$ through self-attention, after which they attend to the transceiver tokens $\mathbf{T}$ through cross-

attention, yielding a scene representation $\mathbf{S}_{\mathrm{c}}$ adapted to the current link and array geometry. The propagation tokens $\mathbf{P}$ are then fed into the RFWeave backbone, where each block first models interactions among them through self-attention and then lets them interact with the concatenated $\mathbf{T}$ and $\mathbf{S}_{\mathrm{c}}$ through cross-attention. Because the queried array orientations and transmit beam are not used in the backbone, its output can be reused across multiple configuration queries for the same scene, transceiver locations, and array geometry.

### 3.4 Spatial Covariance and Power Prediction

**Configuration conditioning.** The backbone representation is shared across configuration queries, while query-specific receiver orientations and the transmit beam are introduced only at the prediction stage. For a queried configuration, let ${}^{G}\mathbf{R}_{T}$ and ${}^{G}\mathbf{R}_{R}$ denote the rotations of TX and RX arrays. The forward and up directions are denoted by $({}^{G}\mathbf{f}_{T}, {}^{G}\mathbf{u}_{T})$ for the TX array and $({}^{G}\mathbf{f}_{R}, {}^{G}\mathbf{u}_{R})$ for the RX array. Let ${}^{T}\mathbf{b}$ denote the transmit beam direction in the TX frame, and let $\mathbf{h}_{i}^{\mathrm{t}}$ and $\mathbf{h}_{j}^{\mathrm{r}}$ denote the final TX- and RX-indexed backbone features. We construct the configuration-conditioned tokens as

$$\begin{aligned}\mathbf{z}_{j}^{\mathrm{r}} &= \mathbf{h}_{j}^{\mathrm{r}} + \phi_{\mathrm{r}}\big({}^{R}\mathbf{p}_{j}/\lambda,\, {}^{G}\mathbf{R}_{R}\,{}^{R}\mathbf{p}_{j}/\lambda,\, {}^{G}\mathbf{f}_{R},\, {}^{G}\mathbf{u}_{R}\big)\,,\\ \mathbf{z}_{i}^{\mathrm{t}} &= \mathbf{h}_{i}^{\mathrm{t}} + \phi_{\mathrm{t}}\big({}^{T}\mathbf{p}_{i}/\lambda,\, {}^{G}\mathbf{R}_{T}\,{}^{T}\mathbf{p}_{i}/\lambda,\, {}^{G}\mathbf{f}_{T},\, {}^{G}\mathbf{u}_{T},\, \mathrm{Re}(w_{i}),\, \mathrm{Im}(w_{i}),\, {}^{G}\mathbf{R}_{T}\,{}^{T}\mathbf{b}\big)\,.\end{aligned} \tag{2}$$

Here, $\phi_{\mathrm{r}}(\cdot)$ and $\phi_{\mathrm{t}}(\cdot)$ are learned projections into the token space, and $w_{i}$ is the complex beamforming weight applied to TX element $i$. The RX conditioning combines local element geometry with the queried array orientation, while the TX conditioning additionally incorporates the applied transmit beam through the beamforming weights and beam direction. The conditioned TX tokens $\mathbf{z}_{i}^{\mathrm{t}}$ are first refined through self-attention. The conditioned RX tokens $\mathbf{z}_{j}^{\mathrm{r}}$ then attend to the refined TX tokens through cross-attention, followed by self-attention.

**Covariance and power decoding.** A shared row-wise factor head maps each final RX token to one row of a complex factor $\mathbf{F} \in \mathbb{C}^{N_{\mathrm{r}} \times r}$, where $r$ is a fixed rank budget. Each row is represented by $2r$ real values corresponding to its real and imaginary parts. Directly predicting the spatial covariance requires $O(N_{\mathrm{r}}^{2})$ outputs without guaranteeing a Hermitian positive semidefinite covariance. We instead normalize the factor as $\overline{\mathbf{F}} = \mathbf{F}/\|\mathbf{F}\|_{F}$ and form $\widehat{\mathbf{R}} = \overline{\mathbf{F}}\,\overline{\mathbf{F}}^{\mathsf{H}}$. This gives a Hermitian positive semidefinite, unit-trace covariance using only $O(N_{\mathrm{r}}r)$ factor outputs. Given a steering vector $\mathbf{a}(\boldsymbol{\omega})$, the AoA spatial spectrum is evaluated as $\widehat{S}(\boldsymbol{\omega}) = 10\log_{10}\Big(\mathbf{a}^{H}(\boldsymbol{\omega})\widehat{\mathbf{R}}\mathbf{a}(\boldsymbol{\omega})\Big)$ and peak-normalized by subtracting its maximum over the queried directions. Predicting $\widehat{\mathbf{R}}$ rather than spectrum values on a fixed angular grid also allows the spectrum to be evaluated at any desired set of steering directions. A separate power head mean-pools the final RX tokens to predict the received power $\widehat{P}$.

### 3.5 Training Objectives

Although the covariance is parameterized through the factor $\mathbf{F}$, we supervise the spatial spectrum induced by the resulting covariance rather than the factor itself. Direct factor supervision is ambiguous because $\mathbf{F}$ and $\mathbf{F}\mathbf{U}$ yield the same covariance for any unitary matrix $\mathbf{U}$. An entry-wise covariance loss is also possible, but it constrains matrix coefficients rather than the directional responses used for evaluation. We instead compute $\widehat{S}$ from $\widehat{\mathbf{R}}$ and supervise the resulting spatial spectrum. This preserves $\widehat{\mathbf{R}}$ as a positive semidefinite, grid-independent output. Because the unit-trace normalization removes the overall covariance scale, the signal level is supervised separately through received power.

We use Smooth-L1 losses $\mathcal{L}_{\mathrm{spec}}$ and $\mathcal{L}_{\mathrm{power}}$ for the peak-normalized AoA spatial spectrum and received power, respectively. We additionally supervise configuration-dependent variation along the receiver orientation and transmit beam axes. Let $S$ denote the target spectrum and $\sigma_{c}(\cdot)$ the per-direction standard deviation across configuration axis $c$. For $c \in \{\mathrm{rx}, \mathrm{beam}\}$, we define $\mathcal{L}_{\mathrm{var},c} = \mathrm{SmoothL1}(\sigma_{c}(\widehat{S}), \sigma_{c}(S))$, averaged over the remaining dimensions. The full training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{spec}} + \mathcal{L}_{\mathrm{power}} + \lambda_{\mathrm{rx}}\mathcal{L}_{\mathrm{var,rx}} + \lambda_{\mathrm{beam}}\mathcal{L}_{\mathrm{var,beam}}. \tag{3}$$

Here, $\lambda_{\mathrm{rx}}$ and $\lambda_{\mathrm{beam}}$ weight the two variation terms, respectively. Each variation term encourages the predicted spectrum to match the target dispersion across the corresponding configuration axis.

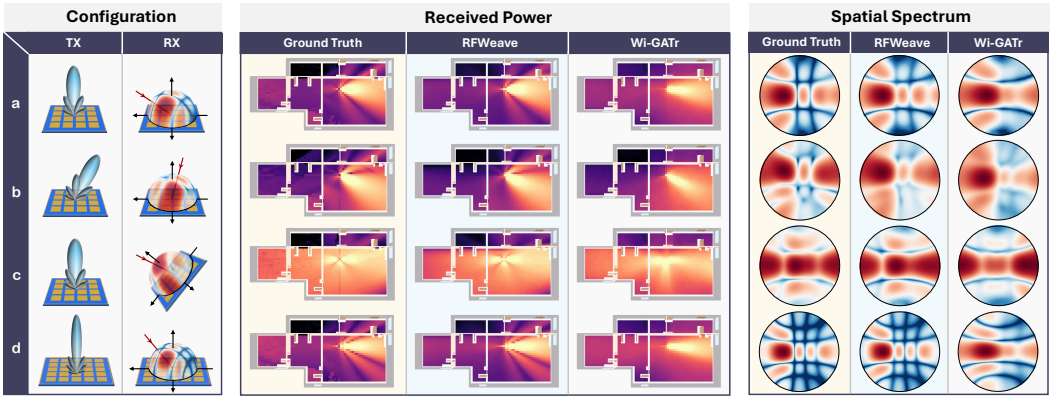


Figure 3: **Comparison on an unseen scene.** We compare received power and spatial spectrum from ground truth, RFWeave, and Wi-GATr under different transceiver configurations. **(a)** uses transmit beam index 0, receiver orientation $180°$, and a $4 \times 4$ UPA. Panels **(b)**–**(d)** vary one configuration factor from this reference: **(b)** changes the transmit beam index to 5, **(c)** changes the receiver orientation to $45°$, and **(d)** increases the array size to $5 \times 5$. Further analysis is provided in Sec. 4.1.

## 4 Experiments

We conduct a series of experiments and ablation studies to answer three questions. (1) How well does RFWeave generalize to unseen scenes under both fixed and varying transceiver configurations? (2) How do key design choices, model capacity, array size, and training data scale affect prediction accuracy and efficiency? (3) Can RFWeave adapt to real-world channel measurements and support downstream beam management? We compare RFWeave against strong baselines in each setting.

**Datasets and configurations.** The main simulated experiments use WiPTR and Wi3R released with Wi-GATr (Hehn et al., 2025). Using far-field synthesis, we construct AoA spatial spectrum and received power targets from their multipath channel records. The resulting data contain 10,000 scenes and over 5 million channel records. Training covers 6 square uniform planar array (UPA) sizes from $3 \times 3$ to $8 \times 8$, 8 receiver orientations at $45°$ intervals, and 16 physical transmit beam directions derived from the $4 \times 4$ UPA DFT codebook. Evaluation further extends to arrays up to $12 \times 12$, 16 unseen receiver orientations at $15°$ intervals, and 9 unseen transmit beam directions. All test scenes in the main generalization experiments are held out from training. Each antenna element follows the directional pattern in 3GPP TR 38.901 (3GPP, 2026). Exact in-distribution (ID) and out-of-distribution (OOD) splits and configuration protocols are provided in Appendix B.

**Baselines and metrics.** We compare RFWeave with PointNet++ (Qi et al., 2017), PointNeXt (Qian et al., 2022), a vanilla Transformer (Vaswani et al., 2017), and Wi-GATr (Hehn et al., 2025). Point cloud baselines use the same material-annotated point clouds as RFWeave, while the Transformer and Wi-GATr follow the original material-annotated meshes. All feed-forward baselines use decoders matched to RFWeave in architecture, parameters, and computation, with an additional array size token encoding the number of antenna elements. For adaptation to target scenes, we further compare with NeRF$^2$ (Zhao et al., 2023) and WRF-GS (Wen et al., 2025). We report spatial spectrum SSIM and MAE, and received power MAE, with higher SSIM and lower MAE indicating better performance.

### 4.1 Experimental Results and Analysis

We consider three RFWeave variants: RFWeave-T, RFWeave-S, and RFWeave-B, corresponding to the Tiny, Small, and Base models, respectively. Their parameter counts and detailed accuracy–efficiency trade-offs are reported in Appendix C.2.

**Cross-scene generalization.** We first isolate scene generalization under a fixed transceiver configuration, with the exact configuration and ID/OOD construction provided in Appendix B.1. As shown in Table 1, RFWeave-B consistently outperforms the cross-scene baselines on both WiPTR

Table 1: Cross-scene propagation prediction on WiPTR and Wi3R. The upper block evaluates feed-forward prediction, while the lower block allows 100 target-scene samples for adaptation.

| Method | Per-scene Opt. | WiPTR | | | | | | Wi3R | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Held-out ID Scenes | | | OOD Scenes | | | Held-out ID Scenes | | | OOD Scenes | | |
| | | SSIM ↑ | $MAE_{Spec}$ ↓ | $MAE_{Pwr}$ ↓ | SSIM ↑ | $MAE_{Spec}$ ↓ | $MAE_{Pwr}$ ↓ | SSIM ↑ | $MAE_{Spec}$ ↓ | $MAE_{Pwr}$ ↓ | SSIM ↑ | $MAE_{Spec}$ ↓ | $MAE_{Pwr}$ ↓ |
| *+ Proposed Decoder* | | | | | | | | | | | | | |
| PointNet++ | None | 0.7988 | 3.16 | 4.75 | 0.7907 | 3.26 | 4.51 | 0.7414 | 4.00 | 8.61 | 0.7073 | 4.43 | 10.91 |
| PointNeXt | None | 0.7993 | 3.16 | 4.34 | 0.7924 | 3.25 | 4.37 | 0.7421 | 3.96 | 8.60 | 0.7092 | 4.38 | 10.87 |
| Wi-GATr | None | 0.7996 | 3.13 | 3.92 | 0.7963 | 3.14 | 4.02 | 0.7371 | 4.03 | 8.62 | 0.5821 | 5.77 | 11.46 |
| Transformer | None | 0.7865 | 3.26 | 4.61 | 0.7803 | 3.36 | 5.14 | 0.7327 | 4.05 | 8.94 | 0.6998 | 4.52 | 11.43 |
| **RFWeave-B** | None | **0.8708** | **2.23** | **2.75** | **0.8597** | **2.34** | **3.39** | **0.7711** | **3.63** | **8.30** | **0.7433** | **4.09** | **10.13** |
| *+ 100 Samples* | | | | | | | | | | | | | |
| $NeRF^2$ | Required | 0.6710 | 3.43 | 1.51 | 0.6408 | 3.71 | 2.62 | 0.7297 | 3.19 | 1.84 | 0.6011 | 4.68 | 4.96 |
| WRF-GS | Required | 0.6397 | 3.83 | 4.52 | 0.6058 | 4.06 | 5.77 | 0.6033 | 5.77 | 4.80 | 0.5846 | 6.14 | 5.52 |
| **RFWeave-B (FT)** | Required | **0.9398** | **1.50** | **0.96** | **0.9211** | **1.76** | **1.97** | **0.9385** | **1.41** | **1.49** | **0.8858** | **2.07** | **2.78** |

Table 2: Single-factor configuration generalization on unseen WiPTR scenes. The upper and lower blocks evaluate seen and unseen configurations, respectively.

| Method | Array Size | | | Receiver Orientation | | | Transmit Beam | | |
|---|---|---|---|---|---|---|---|---|---|
| | SSIM ↑ | $MAE_{Spec}$ ↓ | $MAE_{Pwr}$ ↓ | SSIM ↑ | $MAE_{Spec}$ ↓ | $MAE_{Pwr}$ ↓ | SSIM ↑ | $MAE_{Spec}$ ↓ | $MAE_{Pwr}$ ↓ |
| *Seen configurations, unseen scenes* | | | | | | | | | |
| PointNet++ | 0.7583 | 3.63 | 4.84 | 0.7848 | 3.35 | 4.35 | 0.7521 | 3.71 | 4.36 |
| PointNeXt | 0.7563 | 3.65 | 4.52 | 0.7909 | 3.28 | 4.09 | 0.7520 | 3.69 | 4.34 |
| Wi-GATr | 0.7633 | 3.56 | 3.93 | 0.7843 | 3.34 | 3.58 | 0.7526 | 3.68 | 3.60 |
| Transformer | 0.7411 | 3.84 | 4.71 | 0.7690 | 3.51 | 4.34 | 0.7297 | 3.93 | 4.48 |
| **RFWeave-B$^\dagger$** | 0.8413 | 2.59 | 3.04 | 0.8607 | 2.36 | 2.65 | 0.8473 | 2.54 | 2.73 |
| **RFWeave-B$^\ddagger$** | **0.8453** | **2.56** | **2.82** | **0.8738** | **2.21** | **2.38** | **0.8541** | **2.47** | **2.32** |
| *Unseen configurations, unseen scenes* | | | | | | | | | |
| PointNet++ | 0.6086 | 4.92 | 5.73 | 0.7280 | 3.86 | 4.37 | 0.6893 | 4.33 | 7.61 |
| PointNeXt | 0.6052 | 4.95 | 5.72 | 0.7302 | 3.88 | 4.10 | 0.6989 | 4.28 | 7.38 |
| Wi-GATr | 0.6193 | 4.84 | 5.53 | 0.6906 | 4.26 | 3.64 | 0.7074 | 4.11 | 6.88 |
| Transformer | 0.6112 | 5.07 | 5.79 | 0.7061 | 4.14 | 4.39 | 0.6886 | 4.47 | 7.93 |
| **RFWeave-B$^\dagger$** | 0.6357 | 4.71 | 5.42 | 0.7731 | 3.37 | 2.83 | 0.7213 | 3.99 | 7.12 |
| **RFWeave-B$^\ddagger$** | **0.6459** | **4.65** | **5.40** | **0.7764** | **3.25** | **2.42** | **0.7367** | **3.85** | **6.65** |

and Wi3R. All methods degrade on the OOD scenes, particularly on Wi3R, but RFWeave-B remains the strongest feed-forward model. For comparison with the scene-specific $NeRF^2$ and WRF-GS baselines, RFWeave-B is fine-tuned using the same 100 target-scene samples and improves on every split. Appendix C.1 details the fine-tuning protocol and step sensitivity. Fig. 3 provides a qualitative comparison, where RFWeave better preserves narrow directional power structures and sharp attenuation boundaries while producing a sharper, better-separated spatial spectrum than Wi-GATr.

**Single-factor configuration generalization.** We next vary one configuration factor at a time on unseen WiPTR scenes while keeping other factors fixed. Detailed configuration splits are provided in Appendix B.1. RFWeave-B$^\dagger$ varies only the factor under evaluation during training, whereas RFWeave-B$^\ddagger$ is trained jointly across array sizes, receiver orientations, and transmit beams. Table 2 shows that the two variants remain broadly comparable on seen configurations, while RFWeave-B$^\ddagger$ is consistently stronger under unseen configuration changes. Array size extrapolation produces the largest degradation, while RFWeave-B$^\ddagger$ remains strongest overall. These results indicate that broader configuration training improves transfer without sacrificing accuracy on seen configurations.

**Joint configuration generalization.** Finally, we consider the more challenging case in which all three configuration factors vary jointly. Table 3 evaluates seen configurations, angular interpolation, and array size extrapolation, with their exact construction detailed in Appendix B.1. RFWeave-B achieves the best overall performance in all three regimes. Array size extrapolation is the most challenging setting, with all methods degrading once the array size moves beyond the training range, but RFWeave-B remains strongest overall. Performance also improves consistently from RFWeave-T to RFWeave-B across the three regimes, showing benefits from increased capacity.

Table 3: Joint configuration generalization on unseen WiPTR scenes under seen configurations, angular interpolation, and array size extrapolation.

| Method | Seen Configurations | | | Angular Interpolation | | | Array Size Extrapolation | | |
|---|---|---|---|---|---|---|---|---|---|
| | SSIM ↑ | $\text{MAE}_{\text{Spec}}$ ↓ | $\text{MAE}_{\text{Pwr}}$ ↓ | SSIM ↑ | $\text{MAE}_{\text{Spec}}$ ↓ | $\text{MAE}_{\text{Pwr}}$ ↓ | SSIM ↑ | $\text{MAE}_{\text{Spec}}$ ↓ | $\text{MAE}_{\text{Pwr}}$ ↓ |
| *Unseen scenes* | | | | | | | | | |
| PointNet++ | 0.6952 | 4.41 | 4.74 | 0.6540 | 4.89 | 6.24 | 0.5244 | 6.84 | 5.33 |
| PointNeXt | 0.6968 | 4.40 | 4.55 | 0.6585 | 4.87 | 6.22 | 0.5267 | 6.87 | 5.27 |
| Wi-GATr | 0.6866 | 4.48 | 3.53 | 0.6519 | 4.93 | 5.75 | 0.5396 | 6.75 | 4.93 |
| Transformer | 0.6712 | 4.69 | 4.58 | 0.6444 | 5.01 | 6.01 | 0.5250 | 6.98 | 5.48 |
| **RFWeave-T** | 0.7854 | 3.37 | 2.36 | 0.6987 | 4.42 | 5.59 | 0.5503 | 6.22 | 4.85 |
| **RFWeave-S** | 0.7944 | 3.25 | 2.33 | 0.6992 | 4.41 | 5.52 | 0.5648 | 5.89 | 4.81 |
| **RFWeave-B** | **0.8060** | **3.12** | **2.28** | **0.7082** | **4.32** | **5.45** | **0.5660** | **5.82** | **4.79** |

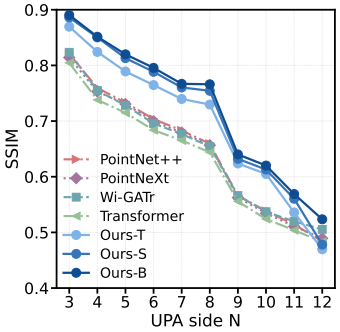


Figure 4: **Effect of array size.** Spatial Spectrum SSIM versus square UPA side length $N$.

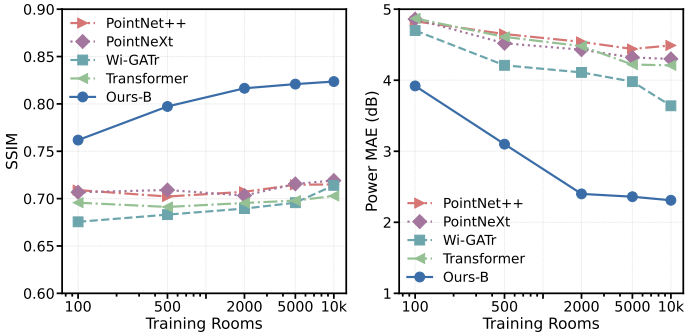


Figure 5: **Data efficiency.** Spatial Spectrum SSIM and received power MAE versus the number of WiPTR training rooms under the same fixed 20k-step optimization budget.

**Effect of array size.** As shown in Fig. 4, we evaluate square UPAs with side length $N = 3, \ldots, 12$. The range $N = 3, \ldots, 8$ is covered during training, while $N = 9, \ldots, 12$ lies beyond the training range. RFWeave-B remains the strongest method in SSIM throughout, but all methods drop noticeably after crossing the training boundary from $N = 8$ to $N = 9$. The power MAE results in Fig. 6 of Appendix C.2 show the same boundary effect. PointNet++ and PointNeXt show a smaller increase in power MAE across this boundary, although their errors are already substantially higher within the training range. Together with Tables 2 and 3, these results show that RFWeave handles in-range array size variation well, while extrapolation to larger unseen arrays remains considerably more difficult.

**Data efficiency.** To examine how performance changes with the amount of training data, we train all methods under the same 20k-step optimization budget using $N_{\text{train}} \in \{100, 500, 2000, 5000, 10000\}$ WiPTR rooms, with array size, receiver orientation, and transmit beam jointly sampled within their training ranges. This isolates the effect of training scene diversity from configuration extrapolation. As shown in Fig. 5, RFWeave-B performs best at every evaluated data scale and improves fastest from 100 to 2,000 training rooms. Beyond 2,000 rooms, the gains diminish as performance gradually saturates. The sampling protocol is detailed in Appendix C.2.

**Model capacity.** We further examine the trade-off between prediction accuracy and computation. RFWeave-T, RFWeave-S, and RFWeave-B contain 4.89M, 10.08M, and 21.67M parameters, respectively. Fig. 8 in Appendix C.2 compares their accuracy and computational cost. Even RFWeave-T outperforms the baselines on both SSIM and received power MAE at lower computational cost, while RFWeave-S and RFWeave-B further improve accuracy at additional cost. Overall, the three variants provide distinct accuracy–computation trade-offs for different resource budgets.

### 4.2 Ablation Studies

**Architectural ablations.** We ablate the main RFWeave design choices in Table 4. Removing configuration conditioning causes the largest degradation, reducing spatial spectrum SSIM by 5.5%, and

Table 4: Ablation of the main RFWeave design choices. $\Delta$ denotes the relative change (%) from the full model. Negative SSIM changes and positive MAE changes indicate degradation.

| Variant | SSIM ↑ | | $\text{MAE}_{\text{Spec}}$ ↓ | | $\text{MAE}_{\text{Pwr}}$ ↓ | |
|---|---|---|---|---|---|---|
| | Value | $\Delta$ (%) | Value | $\Delta$ (%) | Value | $\Delta$ (%) |
| **RFWeave-B (full)** | **0.8060** | – | **3.12** | – | **2.28** | – |
| w/o material properties | 0.7932 | −1.6 | 3.31 | +6.1 | 2.43 | +6.6 |
| w/o pairwise geometry | 0.7756 | −3.8 | 3.48 | +11.5 | 2.67 | +17.1 |
| w/o configuration conditioning | 0.7618 | −5.5 | 3.69 | +18.3 | 2.86 | +25.4 |
| Direct covariance supervision | 0.7827 | −2.9 | 3.41 | +9.3 | 2.59 | +13.6 |

Table 5: Adaptation to measured DICHASUS channels with different numbers of training samples.

| Training samples | **RFWeave-B (FT)** | | | $\text{NeRF}^2$ | | | WRF-GS | | |
|---|---|---|---|---|---|---|---|---|---|
| | SSIM ↑ | $\text{MAE}_{\text{Spec}}$ ↓ | $\text{MAE}_{\text{Pwr}}$ ↓ | SSIM ↑ | $\text{MAE}_{\text{Spec}}$ ↓ | $\text{MAE}_{\text{Pwr}}$ ↓ | SSIM ↑ | $\text{MAE}_{\text{Spec}}$ ↓ | $\text{MAE}_{\text{Pwr}}$ ↓ |
| 100 | **0.7769** | **2.43** | **1.46** | 0.6689 | 2.91 | 2.64 | 0.6428 | 3.76 | 3.03 |
| 500 | **0.7820** | **2.28** | **1.38** | 0.7407 | 2.34 | 1.54 | 0.6433 | 3.53 | 2.76 |
| 1000 | **0.7836** | **2.28** | **1.35** | 0.7443 | 2.34 | 1.47 | 0.6662 | 3.41 | 2.53 |
| 5000 | **0.7921** | **2.27** | 1.31 | 0.7471 | 2.31 | **1.26** | 0.6710 | 3.20 | 2.46 |

increasing received power MAE by 25.4%. Removing pairwise geometry encoding also consistently degrades performance, highlighting the importance of element-level pairwise geometry. Material properties provide a smaller but consistent gain by complementing geometric scene information. Direct covariance supervision degrades all metrics, further supporting spectrum-domain supervision for the directional response of interest.

**Configuration variation supervision.** Each variation term mainly improves its corresponding configuration axis. The balanced setting $(\lambda_{\text{rx}}, \lambda_{\text{beam}}) = (1, 1)$ achieves the best result on five of six metrics and is used in the main experiments. The full sweep is reported in Appendix C.3.

### 4.3 Additional Evaluations

**Measured channel adaptation.** We test simulation-to-measurement adaptation on the single hallway DICHASUS dataset (Euchner et al., 2021). Under the same measured-sample budgets (Table 5), RFWeave achieves the highest spatial spectrum SSIM at every budget and performs best in the low-data regime. At 5,000 samples, $\text{NeRF}^2$ attains slightly lower received power MAE, but RFWeave remains strongest in spatial spectrum prediction.

**Downstream beam management.** By conditioning on transmit beams, RFWeave can be directly applied to downstream beam management. We evaluate a $4 \times 4$ UPA with its DFT codebook in a simulated factory (Bian et al., 2026). RFWeave-B (FT) substantially outperforms location-based steering, increasing mean Top-1 SE from 7.24 to 9.74 bps/Hz (Table 6). Details are provided in Appendix D.

Table 6: Beam management with a $4 \times 4$ DFT codebook. Mean / median spectral efficiency (SE) in bps/Hz. Top-$K$ retains the $K$ highest-ranked beam pairs and reports the best.

| Strategy | Top-1 | Top-2 | Top-3 |
|---|---|---|---|
| Exhaustive sweep (GT) | 14.64 / 14.86 | – | – |
| Location-based steering | 7.24 / 7.31 | – | – |
| **RFWeave-B (FT)** | 9.74 / 10.30 | 10.33 / 11.26 | 10.78 / 11.61 |

## 5 Conclusion

We presented RFWeave, a learned wireless propagation model for joint generalization across scenes, transceiver locations, and transceiver configurations. Variable-length transceiver tokens preserve array structure, while element-indexed propagation tokens encode pairwise geometry between transmitter and receiver elements. Combined with configuration conditioning, these representations enable spatial covariance and received power prediction across array sizes, receiver orientations, and transmit beams. Across WiPTR and Wi3R, RFWeave consistently outperforms learned baselines under scene and configuration shifts. It also adapts to DICHASUS measurements and supports downstream beam management. Together, these results indicate promising progress toward reusable learned propagation modeling across diverse scenes, transceiver locations, and transceiver configurations.

### AI USE STATEMENT

Generative AI tools were used to assist with language polishing, manuscript drafting and revision, literature search, and code development. All technical content, experimental design, implementation, analysis, and conclusions were developed and verified by the authors. The authors take full responsibility for the content of this paper.

### ETHICS STATEMENT

This work studies learned wireless propagation modeling and does not involve human subjects or personally identifiable information. The datasets used in our experiments are publicly available and are used in accordance with their intended research purposes. While improved propagation modeling may benefit wireless communication and autonomous systems, deployment in real-world systems should account for reliability, safety, privacy, and potential misuse.

### REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the RFWeave architecture, training objectives, datasets, transceiver configurations, and evaluation protocols in the main paper and appendix. The appendix further specifies data construction, ID/OOD splits, hyperparameters, fine-tuning procedures, and additional experimental settings. All experiments are based on publicly available WiPTR, Wi3R, and DICHASUS data, with the corresponding preprocessing and far-field synthesis procedures described in the paper. Anonymous source code and data processing scripts are available at https://anonymous.4open.science/r/RFWeave-39B9.

## REFERENCES

3GPP. Study on channel model for frequencies from 0.5 to 100 ghz. Technical Report TR 38.901, 3rd Generation Partnership Project (3GPP), 2026.

Junjie Ai, Shurui Xu, Yanqing Ren, Zhuoyu Liu, Weicong Chen, Wankai Tang, Xiao Li, Chao-Kai Wen, and Shi Jin. Propagation-consistent wireless environment digital twin construction under sparse measurements, 2026. arXiv:2605.22361.

Muhammad Alrabeiah and Ahmed Alkhateeb. Deep learning for mmwave beam and blockage prediction using sub-6 ghz channels. *IEEE Transactions on Communications*, 68(9):5504–5518, 2020. doi: 10.1109/TCOMM.2020.3003670.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Yijie Bian, Wei Guo, Zixin Wang, Shenghui Song, Jun Zhang, and Khaled B. Letaief. GSBF: Gaussian splatting for environment-aware beamforming, 2026. arXiv:2608.05896.

Benedikt Böck, Andreas Oeldemann, Timo Mayer, Francesco Rossetto, and Wolfgang Utschick. Physics-informed generative modeling of wireless channels. In *International Conference on Machine Learning (ICML)*, pp. 4602–4626. PMLR, 2025.

Johann Brehmer, Pim de Haan, Sönke Behrends, and Taco S. Cohen. Geometric algebra transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Gouranga Charan, Umut Demirhan, João Morais, Arash Behboodi, Hamed Pezeshki, and Ahmed Alkhateeb. Multi-modal beam prediction challenge 2022: Towards generalization, 2022. arXiv:2209.07519.

Xingyu Chen, Zihao Feng, Kun Qian, and Xinyu Zhang. Radio frequency ray tracing with neural object representation for enhanced rf modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21339–21348, 2025. doi: 10.1109/CVPR52734.2025.01988.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Florian Euchner, Marc Gauger, Sebastian Dörner, and Stephan ten Brink. A distributed massive mimo channel sounder for "big csi data"-driven machine learning. In *WSA 2021; 25th International ITG Workshop on Smart Antennas*, pp. 1–6, 2021.

Thomas Hehn, Markus Peschl, Tribhuvanesh Orekondy, Arash Behboodi, and Johann Brehmer. Differentiable and learnable wireless simulation with geometric transformers. In *International Conference on Learning Representations (ICLR)*, 2025.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Ibrahim Kilinc and Robert W. Heath. Heterogeneity-agnostic ai/ml-assisted beam selection for multi-panel arrays. *IEEE Transactions on Machine Learning in Communications and Networking*, pp. 1318–1334, 2026. doi: 10.1109/TMLCN.2026.3715130.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Aldebaro Klautau, Nuria González-Prelcic, and Robert W. Heath. LIDAR data for deep learning-based mmwave beam-selection. *IEEE Wireless Communications Letters*, 8(3):909–912, 2019. doi: 10.1109/LWC.2019.2899571.

Haofan Lu, Christopher Vattheuer, Baharan Mirzasoleiman, and Omid Abari. NeWRF: A deep learning framework for wireless radiation field reconstruction and channel prediction. In *International Conference on Machine Learning (ICML)*, pp. 33147–33159. PMLR, 2024.

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 *International Conference on Learning Representations (ICLR)*, 2023.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Sajad Rezaie, Elisabeth de Carvalho, and Carles Navarro Manchón. A deep learning approach to location- and orientation-aided 3d beam selection for mmwave communications. *IEEE Transactions on Wireless Communications*, 21(12):11110–11124, 2022a. doi: 10.1109/TWC.2022.3189788.

Sajad Rezaie, João Morais, Ahmed Alkhateeb, and Carles Navarro Manchón. Device-agnostic millimeter wave beam selection using machine learning, 2022b. arXiv:2211.12125.

Sajad Rezaie, João Morais, Elisabeth de Carvalho, Ahmed Alkhateeb, and Carles Navarro Manchón. Location- and orientation-aware millimeter wave beam selection for multi-panel antenna devices. In *IEEE Global Communications Conference (GLOBECOM)*, pp. 597–602, 2022c. doi: 10.1109/GLOBECOM48099.2022.10001089.

Noam Shazeer. GLU variants improve transformer, 2020. arXiv:2002.05202.

Muhammad Umer, Muhammad Ahmed Mohsin, Ahsan Bilal, and John M. Cioffi. Neural gaussian radio fields for channel estimation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 12138–12149, 2026. doi: 10.1145/3770855.3818923.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Xiucheng Wang, Keda Tao, Nan Cheng, Zhisheng Yin, Zan Li, Yuan Zhang, and Xuemin Shen. RadioDiff: An effective generative diffusion model for sampling-free dynamic radio map construction. *IEEE Transactions on Cognitive Communications and Networking*, 11(2):738–750, 2025. doi: 10.1109/TCCN.2024.3504489.

Chaozheng Wen, Jingwen Tong, Yingdong Hu, Zehong Lin, and Jun Zhang. WRF-GS: Wireless radiation field reconstruction with 3d gaussian splatting. In *IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–10, 2025. doi: 10.1109/INFOCOM55648.2025.11044513.

Chaozheng Wen, Chenghong Bian, Hongze Chen, and Jun Zhang. Point2Radio: A foundation model for cross-scene radio fields from material-aware point clouds, 2026. arXiv:2607.28994.

Weihua Xu, Feifei Gao, Shi Jin, and Ahmed Alkhateeb. 3d scene-based beam selection for mmwave communications. *IEEE Wireless Communications Letters*, 9(11):1850–1854, 2020. doi: 10.1109/LWC.2020.3005983.

Kang Yang, Yuning Chen, and Wan Du. Generalizable radio-frequency radiance fields for spatial spectrum synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12533–12543, 2026.

Tianyu Yang, Mahdi Barzegar Khalilsarai, Saeid Haghighatshoar, and Giuseppe Caire. Structured channel covariance estimation from limited samples for large antenna arrays. *EURASIP Journal on Wireless Communications and Networking*, 2023:24, 2023. doi: 10.1186/s13638-023-02235-6.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Xiaopeng Zhao, Zhenlin An, Qingrui Pan, and Lei Yang. NeRF2: Neural radio-frequency radiance fields. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2023. doi: 10.1145/3570361.3592527.

# A Implementation Details

## A.1 Details of Scene Tokenizer.

We provide additional implementation details for the scene tokenizer introduced in Sec. 3.2. Each input point is parameterized by its global position ${}^{G}\mathbf{p}$, surface normal ${}^{G}\mathbf{n}$, and material properties $\boldsymbol{\mu} = [\epsilon_r, \log_{10}\sigma]$, where $\epsilon_r$ and $\sigma$ denote relative permittivity and conductivity, respectively. The two material channels are standardized using statistics from the corresponding training split and encoded by a two-layer MLP. The resulting material embedding is then fused with the geometric features. At each set abstraction stage, the sampled center feature serves as the attention query. Neighboring features are augmented with their relative displacements and Euclidean distances to the center before forming the keys and values. The three stages use neighborhoods of 24, 32, and 64 points, respectively, with a 64-dimensional local representation at each stage. The resulting local representations are combined with a scene one-hot type vector and projected to the common token dimension $d$, forming the material-aware scene tokens $\mathbf{S}$. Since scene tokenization depends only on the environment, the scene tokens are computed once per scene and reused across all queried transceiver links. Table 7 summarizes the scene and transceiver tokenization used in RFWeave.

## A.2 Details of Transceiver Tokenizer.

The transceiver tokenizer represents transceiver geometry at the element, array, and link levels. To distinguish different token roles, we use six one-hot type vectors $\mathbf{t}_{\mathrm{te}}, \mathbf{t}_{\mathrm{re}}, \mathbf{t}_{\mathrm{tc}}, \mathbf{t}_{\mathrm{rc}}, \mathbf{t}_{\mathrm{link}}, \mathbf{t}_{\mathrm{scene}} \in \{0,1\}^6$, corresponding to TX elements, RX elements, TX centers, RX centers, TX–RX links, and scene tokens, respectively. A shared learned projection $\phi(\cdot)$ maps the geometric features and type vectors to the common $d$-dimensional token space. For carrier wavelength $\lambda$, the TX and RX element tokens are given by $\mathbf{x}_i^{\mathrm{t}} = \phi({}^{T}\mathbf{p}_i/\lambda, \mathbf{t}_{\mathrm{te}})$ and $\mathbf{x}_j^{\mathrm{r}} = \phi({}^{R}\mathbf{p}_j/\lambda, \mathbf{t}_{\mathrm{re}})$, where the element positions are expressed in their respective local array frames. Following the TX–RX link tokenization in Wi-GATr (Hehn et al., 2025), the array centers are represented by $\mathbf{x}_c^{\mathrm{t}} = \phi({}^{G}\mathbf{c}^{\mathrm{t}}, \mathbf{t}_{\mathrm{tc}})$ and $\mathbf{x}_c^{\mathrm{r}} = \phi({}^{G}\mathbf{c}^{\mathrm{r}}, \mathbf{t}_{\mathrm{rc}})$. The link token is defined as $\mathbf{x}_{\mathrm{link}} = \phi({}^{G}\boldsymbol{\ell}, \mathbf{t}_{\mathrm{link}})$, where ${}^{G}\boldsymbol{\ell} = {}^{G}\mathbf{c}^{\mathrm{r}} - {}^{G}\mathbf{c}^{\mathrm{t}}$ denotes the displacement from the TX array center to the RX array center in the global frame. Together, these tokens form the transceiver sequence $\mathbf{T} = [\mathbf{x}_{1:N_{\mathrm{t}}}^{\mathrm{t}}, \mathbf{x}_{1:N_{\mathrm{r}}}^{\mathrm{r}}, \mathbf{x}_c^{\mathrm{t}}, \mathbf{x}_c^{\mathrm{r}}, \mathbf{x}_{\mathrm{link}}] \in \mathbb{R}^{(N_{\mathrm{t}}+N_{\mathrm{r}}+3)\times d}$. Its length varies naturally with $N_{\mathrm{t}}$ and $N_{\mathrm{r}}$, allowing the same tokenizer to accommodate different array sizes.

Table 7: Scene and transceiver tokenization. Token roles are distinguished by one-hot vectors.

| Data type | Input parameterization | Tokenization | Type vector |
|---|---|---|---|
| Scene | ${}^{G}\mathbf{p}$, ${}^{G}\mathbf{n}$, $\boldsymbol{\mu}$ | 1 token per final sampled center | $\mathbf{t}_{\mathrm{scene}}$ |
| TX/RX elements | ${}^{T}\mathbf{p}_i/\lambda$, ${}^{R}\mathbf{p}_j/\lambda$ | 1 token per antenna element | $\mathbf{t}_{\mathrm{te}}$, $\mathbf{t}_{\mathrm{re}}$ |
| Array centers | ${}^{G}\mathbf{c}^{\mathrm{t}}$, ${}^{G}\mathbf{c}^{\mathrm{r}}$ | 1 token per array | $\mathbf{t}_{\mathrm{tc}}$, $\mathbf{t}_{\mathrm{rc}}$ |
| TX–RX link | ${}^{G}\boldsymbol{\ell}$ | 1 token per link | $\mathbf{t}_{\mathrm{link}}$ |

## A.3 Details of Backbone.

**Bidirectional pairwise aggregation.** The pairwise tensor $\mathbf{E}$ is constructed as described in Sec. 3.3. Before aggregating the pairwise features, the TX center token $\mathbf{x}_c^{\mathrm{t}}$ and RX center token $\mathbf{x}_c^{\mathrm{r}}$ serve as queries and attend to their respective element tokens $\mathbf{x}_{1:N_{\mathrm{t}}}^{\mathrm{t}}$ and $\mathbf{x}_{1:N_{\mathrm{r}}}^{\mathrm{r}}$, thereby summarizing the overall structure of each array. These summaries are then used to construct array-indexed queries. Specifically, the TX-indexed query $\mathbf{q}_i^{\mathrm{t}}$ combines the corresponding TX element token $\mathbf{x}_i^{\mathrm{t}}$, the link token $\mathbf{x}_{\mathrm{link}}$, and the RX array summary, while the RX-indexed query $\mathbf{q}_j^{\mathrm{r}}$ is constructed symmetrically from $\mathbf{x}_j^{\mathrm{r}}$, $\mathbf{x}_{\mathrm{link}}$, and the TX array summary. Learned attention aggregators $\mathcal{A}_{\mathrm{t}}$ and $\mathcal{A}_{\mathrm{r}}$ then summarize each row and column of $\mathbf{E}$ as $\mathbf{z}_i^{\mathrm{t}} = \mathcal{A}_{\mathrm{t}}(\mathbf{q}_i^{\mathrm{t}}, \mathbf{E}_{i,:})$ and $\mathbf{z}_j^{\mathrm{r}} = \mathcal{A}_{\mathrm{r}}(\mathbf{q}_j^{\mathrm{r}}, \mathbf{E}_{:,j})$. The resulting propagation tokens are concatenated as $\mathbf{P} = \mathrm{concat}(\mathbf{z}_{1:N_{\mathrm{t}}}^{\mathrm{t}}, \mathbf{z}_{1:N_{\mathrm{r}}}^{\mathrm{r}})$. Each pairwise feature contributes through both a TX-indexed row and an RX-indexed column. For square $n \times n$ arrays at both ends, $\mathbf{E}$ contains $n^4$ pairwise features, whereas the backbone processes only $2n^2$ propagation tokens.

**Architecture and model variants.** The backbone uses two self-attention blocks for scene refinement. Propagation tokens are refined through a sequence of stages. Each stage applies self-attention to the propagation tokens, followed by cross-attention to the transceiver and scene context. The scene

conditioning and propagation refinement blocks use a pre-normalization design with RMSNorm (Zhang & Sennrich, 2019), while QK normalization (Henry et al., 2020) is applied to the query and key features before attention computation (Dao et al., 2022). All variants use multi-head attention with eight heads. Each block further includes a SwiGLU feed-forward layer (Shazeer, 2020) with an equivalent FFN expansion ratio of 4. RFWeave-T, RFWeave-S, and RFWeave-B share the same scene and transceiver tokenizers, backbone structure, configuration conditioning, and prediction interface. Model capacity is varied through the hidden dimension and the number of propagation refinement stages, while the decoder width is scaled accordingly. Table 8 summarizes the model settings for the three RFWeave variants.

Table 8: Architecture and parameter counts of the RFWeave variants. Scene self-attention uses two blocks and the covariance factor rank is $r = 16$ for all variants.

| Model | Hidden dim. | Heads | Prop. stages | Backbone (M) | Decoder (M) | Total (M) |
|---|---|---|---|---|---|---|
| RFWeave-T | 128 | 8 | 6 | 3.62 | 1.27 | 4.89 |
| RFWeave-S | 192 | 8 | 6 | 7.38 | 2.70 | 10.08 |
| RFWeave-B | 256 | 8 | 8 | 16.63 | 5.04 | 21.67 |

# B Experimental Details

## B.1 Datasets and Evaluation Protocols.

**Dataset construction.** The main simulated experiments use the path-level channel records released with WiPTR (Hehn et al., 2025) and Wi3R (Orekondy et al., 2023), both generated at 3.5 GHz. We apply deterministic far-field synthesis to each record to obtain the normalized receive-side spatial covariance, AoA spatial spectrum, and received power for a given transceiver configuration. This allows propagation targets for different array sizes, receiver orientations, and transmit beams to be generated from the same record without rerunning the ray tracer. All square UPAs use half-wavelength element spacing. The 16 training transmit beams are defined by the physical directions of a $4 \times 4$ UPA DFT codebook, while the beamforming vectors for other array sizes are adjusted to preserve these directions. Table 9 summarizes the configuration protocols used for training and evaluation.

Table 9: Training and evaluation configurations for generalization experiments. *Seen* denotes values observed during training, *unseen* denotes held-out values used for generalization evaluation, and *fixed* denotes the reference value held constant in the corresponding single-factor experiment.

| Regime | Array size | Receiver orientation | Transmit beam |
|---|---|---|---|
| Training | 6 seen sizes ($3 \times 3$–$8 \times 8$) | 8 seen orientations | 16 seen directions |
| Single-factor: array size | 4 unseen sizes ($9 \times 9$–$12 \times 12$) | fixed | fixed |
| Single-factor: receiver orientation | fixed | 16 unseen orientations | fixed |
| Single-factor: transmit beam | fixed | fixed | 9 unseen directions |
| Joint: seen | 6 seen sizes ($3 \times 3$–$8 \times 8$) | 8 seen orientations | 16 seen directions |
| Joint: angular interpolation | 6 seen sizes ($3 \times 3$–$8 \times 8$) | 8 seen orientations | 9 unseen directions |
| Joint: array size extrapolation | 4 unseen sizes ($9 \times 9$–$12 \times 12$) | 8 seen orientations | 16 seen directions |

**Cross-scene generalization protocol.** We evaluate cross-scene generalization under a fixed transceiver configuration to isolate scene variation. Both TX and RX use a $4 \times 4$ square UPA. The transmitter uses transmit beam index 0, and the TX and RX orientations are aligned with the global $+x$ and $-x$ axes, respectively. All evaluation scenes are held out from training. For WiPTR, the ID scenes follow the same generation distribution as the training scenes, whereas the OOD scenes alter room connectivity through wall removal. For Wi3R, the ID scenes contain three rooms, whereas the OOD scenes contain four rooms.

**Single-factor configuration generalization protocol.** We evaluate single-factor generalization on unseen WiPTR scenes following the protocols in Table 9. RFWeave-B$^{\dagger}$ varies only the factor under evaluation during training, while the other two factors remain fixed. RFWeave-B$^{\ddagger}$, in contrast, is trained jointly across array sizes, receiver orientations, and transmit beams. The two variants use the

same number of training samples for every complete configuration they share, while RFWeave-B$^{\ddagger}$ additionally covers a broader set of distinct configurations.

**Joint configuration generalization protocol.** All joint generalization regimes use the same held-out WiPTR scenes, with configuration factors sampled according to Table 9. The scene split is fixed across the three regimes, while the queried configurations cover seen values within the training ranges, unseen transmit beam directions interpolated within the training range, and unseen array sizes extrapolated beyond the training range.

### B.2 Baselines and Optimization.

**Baseline implementations.** All reproduced feed-forward baselines use the same data splits, optimization budget, and evaluation protocol as RFWeave. PointNet++ (Qi et al., 2017) and PointNeXt (Qian et al., 2022) receive the same material-annotated point clouds as RFWeave, while Wi-GATr (Hehn et al., 2025) and the vanilla Transformer (Vaswani et al., 2017) follow the material-annotated mesh representation adopted in the original Wi-GATr. All four baselines use the same covariance and power prediction interface as RFWeave, with matched decoder capacity. To accommodate different array sizes, we append an additional array size token to each baseline to encode the number of antenna elements.

**PointNet++ and PointNeXt.** PointNet++ uses a set abstraction encoder with the same point cloud preprocessing and material attributes as RFWeave. PointNeXt follows the official PointNeXt-B configuration. Its input head has width 32, while the four set abstraction stages use stride 4 with output widths of 64, 128, 256, and 512. The corresponding block counts are 2, 3, 2, and 2. Both methods use 3D coordinates for point set geometry operations, while surface normals, relative permittivity, and conductivity are provided as point features.

**Wi-GATr.** Our reproduction follows the predictive model setting of Hehn et al. (2025). It uses 32 geometric-transformer blocks (Brehmer et al., 2023) with 16 hidden multivector channels, 32 hidden scalar channels, and 8 attention heads. The link, TX, RX, and mesh-face tokens follow the geometric roles defined in the original tokenizer. Scalar channels encode token type and material identity.

**Transformer.** The vanilla Transformer uses the same mesh-based input representation as Wi-GATr but removes the geometric-algebra equivariant embedding. It contains 32 Transformer blocks with a hidden dimension of 96 and 8 attention heads. Each block applies multi-head self-attention followed by a feed-forward network. The feed-forward network uses an MLP ratio of 4 and GELU activations (Hendrycks & Gimpel, 2016). LayerNorm (Ba et al., 2016) is applied within each block.

**Scene-specific baselines.** NeRF$^2$ (Zhao et al., 2023) and WRF-GS (Wen et al., 2025) are reconstruction methods optimized independently for each target scene. In the adaptation comparison, both methods use the same 100 target-scene samples provided to RFWeave-B for fine-tuning. RFWeave-B, in contrast, starts from its pretrained checkpoint and adapts to the target scene.

**Optimization and adaptation.** All feed-forward models are trained with Adam (Kingma & Ba, 2015) for 20k steps. The learning rate is selected separately for each model by grid search, with $10^{-3}$ yielding the best validation performance across all reported models. Each training step samples three scenes and 32 transceiver links per scene, giving 96 links in total while allowing links from the same scene to share the scene representation. All reproduced results are obtained on NVIDIA H100 GPUs. For target-scene adaptation, RFWeave-B is initialized from the pretrained feed-forward checkpoint and fine-tuned for 500 steps using 100 target-scene samples. NeRF$^2$ and WRF-GS are optimized independently for each target scene for 20k and 60k steps, respectively.

## C Additional Experiments

### C.1 Fine-Tuning Sensitivity

To examine whether the adaptation performance of RFWeave-B depends strongly on the fine-tuning duration, we vary the number of fine-tuning steps while keeping the target-scene sample budget fixed

at 100. Table 10 reports the results with 500, 1,000, and 2,000 fine-tuning steps for the scene-specific comparison in Table 1. Increasing the number of steps does not lead to consistent improvements across the reported metrics. We therefore use 500 fine-tuning steps in the main experiments.

Table 10: Fine-tuning sensitivity to the number of optimization steps. All settings use the same 100 target-scene samples and the fine-tuning configuration adopted in Table 1.

| Steps | WiPTR Held-out ID Scenes | | | Wi3R OOD Scenes | | |
|---|---|---|---|---|---|---|
| | SSIM ↑ | $\text{MAE}_{\text{Spec}}$ ↓ | $\text{MAE}_{\text{Pwr}}$ ↓ | SSIM ↑ | $\text{MAE}_{\text{Spec}}$ ↓ | $\text{MAE}_{\text{Pwr}}$ ↓ |
| 500 | 0.9398 | **1.498** | **0.962** | **0.8858** | **2.069** | 2.782 |
| 1000 | 0.9372 | 1.576 | 0.970 | 0.8828 | 2.113 | **2.115** |
| 2000 | **0.9407** | 1.556 | 1.045 | 0.8603 | 2.438 | 3.010 |

## C.2 Scaling and Efficiency

For the data efficiency study in Fig. 5, each method is trained for 20k steps using $N_{\text{train}} \in \{100, 500, 2000, 5000, 10000\}$ WiPTR scenes. During training, array size, receiver orientation, and transmit beam are sampled jointly within their respective training ranges. All models are evaluated on the same 2,000 deterministically sampled transceiver links from 100 held-out scenes.

For the model capacity and array size studies, the scene representation is computed once, after which a single transceiver link is decoded for all 16 transmit beam directions and 8 receiver orientations. The spatial spectrum is computed on the same $360 \times 90$ angular grid used throughout evaluation. Total GFLOPs account for scene and backbone processing, prediction decoding, and spatial spectrum computation. Peak GPU memory is measured with 8 transceiver links sharing a single scene representation. Table 11 reports the total forward computation for all feed-forward models at $4 \times 4$ and $8 \times 8$ array sizes.

Table 11: Model size and total forward computation at $4 \times 4$ and $8 \times 8$ array sizes.

| Method | Params. (M) | $4 \times 4$ GFLOPs | $8 \times 8$ GFLOPs |
|---|---|---|---|
| PointNet++ | 5.19 | 21.05 | 67.49 |
| PointNeXt | 8.52 | 28.53 | 74.97 |
| Wi-GATr | 6.47 | 37.73 | 84.16 |
| Transformer | 8.22 | 21.50 | 67.93 |
| RFWeave-T | 4.89 | 10.36 | 28.99 |
| RFWeave-S | 10.08 | 16.80 | 47.90 |
| RFWeave-B | 21.67 | 27.47 | 79.80 |

The additional scaling experiments reveal how prediction accuracy and computational requirements change with array size and model capacity. For received power prediction in Fig. 6, RFWeave-S

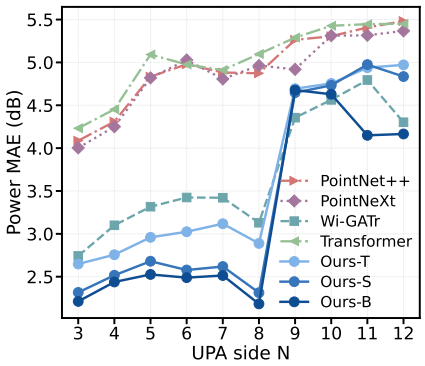


Figure 6: **Received power MAE across array sizes.** Models are trained on square UPAs with $N = 3, \ldots, 8$ and additionally evaluated on unseen array sizes $N = 9, \ldots, 12$.

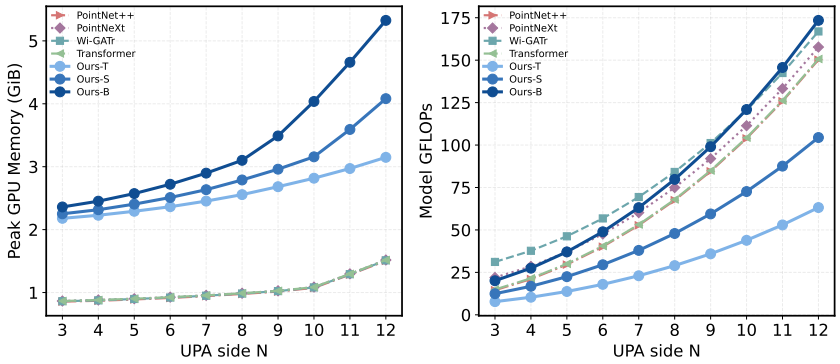


Figure 7: **Computational scaling.** Peak GPU memory is shown on the left and total forward GFLOPs on the right for square UPAs with varying side length $N$.

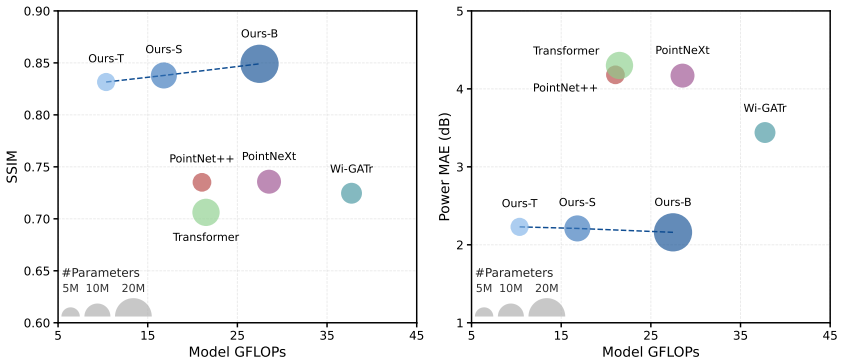


Figure 8: **Accuracy–efficiency trade-off.** Spatial spectrum SSIM is shown on the left and received power MAE on the right against total forward GFLOPs. Bubble size indicates the number of model parameters.

and RFWeave-B maintain substantially lower MAE than the feed-forward baselines throughout the training range of $N = 3, \ldots, 8$, with RFWeave-B remaining around 2.2–2.6 dB. At the first unseen array size ($N = 9$), Wi-GATr and the RFWeave variants show a clear increase in received power MAE. The corresponding degradation in spatial spectrum SSIM in Fig. 4 is much milder. This suggests that received power prediction is more sensitive to array size extrapolation, while the spatial structure of the spectrum remains relatively stable. The computational results in Fig. 7 show that RFWeave-T and RFWeave-S require substantially fewer GFLOPs than the feed-forward baselines across the evaluated array sizes, while the computational cost of RFWeave-B remains comparable to several baselines. Peak GPU memory increases with array size for all models. The increase becomes more pronounced for the larger RFWeave variants, with RFWeave-B requiring the most memory at large array sizes. The accuracy–computation comparison in Fig. 8 further shows that RFWeave-T already achieves an SSIM above 0.83 and a received power MAE of approximately 2.2 dB at only about 10 GFLOPs. It outperforms all feed-forward baselines on both metrics while requiring less computation than most of them. RFWeave-S and RFWeave-B further improve both metrics, although the gains become smaller as model capacity increases. Overall, the three variants provide different accuracy–efficiency trade-offs. RFWeave-T achieves strong accuracy at relatively low computational cost, whereas the larger variants offer further improvements at higher computation and memory cost.

### C.3 Configuration Variation Loss

To examine the weight setting of the configuration variation loss, we conduct an ablation using 2,000 WiPTR training rooms with a fixed training budget of 20k steps. Each setting is evaluated on 100 unseen rooms, where we test generalization to unseen receiver orientations and unseen transmit

beam directions. We vary the weights of the two variation terms while keeping all other training settings unchanged. Table 12 reports the full sweep. The receiver orientation term mainly improves generalization across receiver orientations, while the transmit beam term mainly benefits unseen beam directions. The balanced setting $(\lambda_{\mathrm{rx}}, \lambda_{\mathrm{beam}}) = (1, 1)$ achieves the best performance on five of the six reported metrics and provides the strongest overall results. We therefore use this setting in the main experiments.

Table 12: Sensitivity to the configuration variation loss weights.

| $\lambda_{\mathrm{rx}}$ | $\lambda_{\mathrm{beam}}$ | Unseen RX Orientation | | | Unseen TX Beam | | |
|---|---|---|---|---|---|---|---|
| | | SSIM ↑ | $\mathrm{MAE}_{\mathrm{Spec}}$ ↓ | $\mathrm{MAE}_{\mathrm{Pwr}}$ ↓ | SSIM ↑ | $\mathrm{MAE}_{\mathrm{Spec}}$ ↓ | $\mathrm{MAE}_{\mathrm{Pwr}}$ ↓ |
| 0 | 0 | 0.7128 | 3.809 | 2.692 | 0.7205 | 4.051 | 6.789 |
| 1 | 0 | 0.7485 | 3.525 | 2.771 | 0.7260 | 3.965 | 6.622 |
| 2 | 0 | 0.7471 | 3.513 | 2.785 | 0.7144 | 4.094 | 6.911 |
| 0 | 1 | 0.7138 | 3.876 | 3.052 | **0.7472** | 4.045 | 7.197 |
| 0 | 2 | 0.7171 | 3.812 | 2.879 | 0.7349 | 3.955 | 6.845 |
| 0.5 | 0.5 | 0.6880 | 4.103 | 3.081 | 0.7230 | 4.096 | 6.911 |
| **1** | **1** | **0.7533** | **3.486** | **2.253** | 0.7402 | **3.870** | **6.562** |
| 1.5 | 1.5 | 0.7431 | 3.600 | 2.798 | 0.7301 | 4.042 | 6.812 |
| 2 | 2 | 0.7466 | 3.538 | 2.989 | 0.7276 | 4.079 | 6.850 |

## C.4 Qualitative Comparisons

Fig. 9–Fig. 12 provide additional qualitative comparisons for cross-scene prediction and single-factor configuration generalization. The first example additionally includes target-scene adaptation, while the remaining examples show feed-forward predictions when one transceiver configuration factor varies at a time.

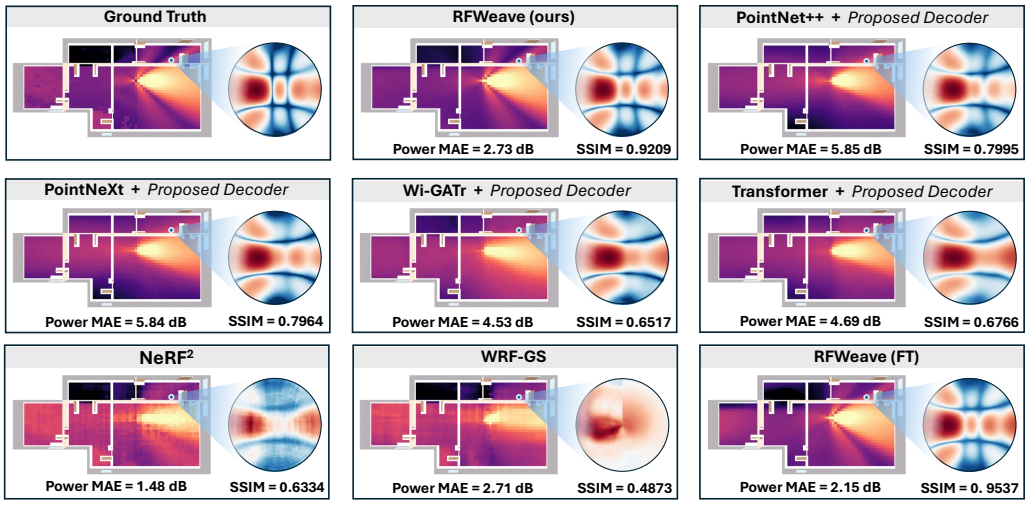


Figure 9: **Cross-scene prediction and target-scene adaptation.** Qualitative comparison on an unseen scene. Each method predicts the received power map and the AoA spatial spectrum at the marked receiver location. RFWeave denotes the feed-forward prediction, while RFWeave (FT) is fine-tuned using 100 target-scene samples. NeRF$^2$ and WRF-GS are optimized independently for the target scene using the same samples. RFWeave better preserves the narrow high-power structure and separated spectrum lobes than the feed-forward baselines.

# D Downstream Beam Management

**Evaluation setup.** We further evaluate whether RFWeave can support downstream beam decisions beyond propagation prediction itself (Rezaie et al., 2022b; Charan et al., 2022; Klautau et al., 2019; Alrabeiah & Alkhateeb, 2020). We use the simulated factory environment from Bian et al. (2026).

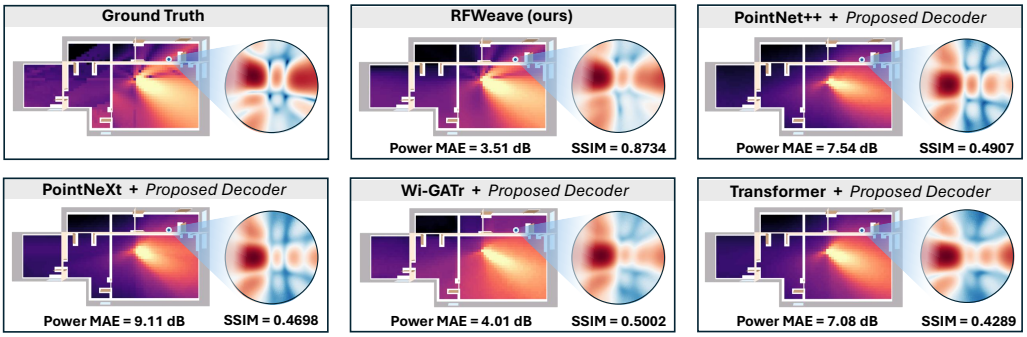


Figure 10: **Transmit beam generalization.** Feed-forward predictions on an unseen WiPTR scene as the transmit beam changes from index 0 to 5, while array size and receiver orientation remain fixed.

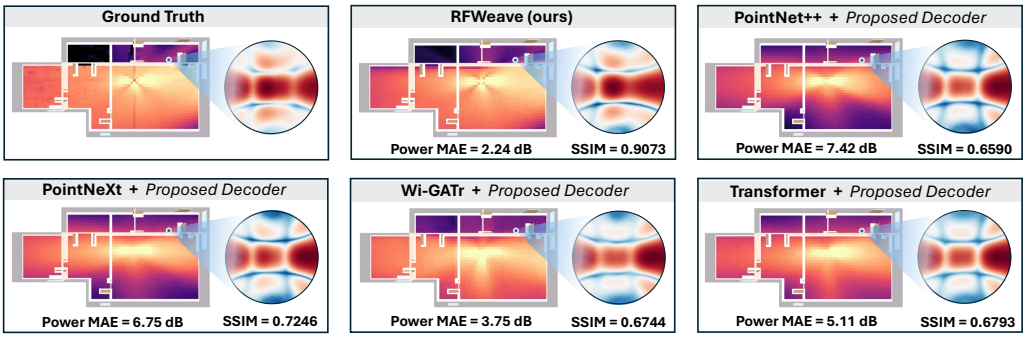


Figure 11: **Receiver orientation generalization.** Feed-forward predictions on an unseen WiPTR scene as the receiver orientation changes from the reference orientation to $45°$, while array size and transmit beam remain fixed.

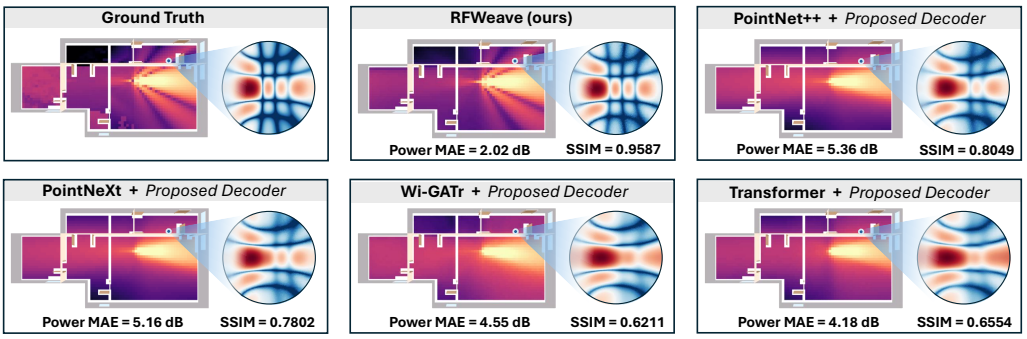


Figure 12: **Array size generalization.** Feed-forward predictions on an unseen WiPTR scene as the square UPA changes from $4 \times 4$ to $5 \times 5$, while receiver orientation and transmit beam remain fixed.

Both the transmitter and receiver employ a $4 \times 4$ UPA, and beam selection is performed over a 16-beam DFT codebook. RFWeave-B is initialized from the pretrained checkpoint and then adapted to the factory environment. The factory scene contains five material classes: wood, concrete, metal, chipboard, and marble. These materials are represented using the same continuous material encoding and dataset-specific normalization adopted by the scene tokenizer.

**Beam-pair scoring.** Let $\{\boldsymbol{\omega}_q\}_{q=1}^{Q}$ denote the AoA steering directions on the angular evaluation grid, where $q$ indexes the individual directions. For each transmit beam index $b_t$, RFWeave predicts a peak-normalized AoA spatial spectrum $\widehat{S}_{\mathrm{dB}}(b_t, \boldsymbol{\omega}_q)$ over these directions, together with a received

power estimate $\widehat{P}_{\mathrm{dB}}(b_t)$. We convert the predicted spectrum from dB to normalized linear spectrum weights,

$$\widehat{s}(b_t, q) = \frac{10^{\widehat{S}_{\mathrm{dB}}(b_t, \boldsymbol{\omega}_q)/10}}{\sum_{q'=1}^{Q} 10^{\widehat{S}_{\mathrm{dB}}(b_t, \boldsymbol{\omega}_{q'})/10}}, \qquad \sum_{q=1}^{Q} \widehat{s}(b_t, q) = 1. \tag{4}$$

Let $\mathbf{a}_r(\boldsymbol{\omega}_q)$ denote the normalized RX steering vector for direction $\boldsymbol{\omega}_q$, and let $\mathbf{w}_r^{(b_r)}$ denote the RX codebook vector for beam index $b_r$. We define a score for each TX/RX beam pair as

$$\widehat{G}(b_t, b_r) = 10^{\widehat{P}_{\mathrm{dB}}(b_t)/10} \sum_{q=1}^{Q} \widehat{s}(b_t, q) \left| \mathbf{a}_r(\boldsymbol{\omega}_q)^{\mathsf{H}} \mathbf{w}_r^{(b_r)} \right|^2. \tag{5}$$

This spectrum-based projection combines the relative directional structure encoded by the predicted spatial spectrum with the received power estimate to form a beam-pair ranking score. The resulting score serves to rank candidate beam pairs rather than to directly estimate physical beam-pair power.

Ground-truth beam-pair power is computed directly from the multipath record. Let $g_l$ denote the power of path $l$, and let $\Omega_l^{\mathrm{t}}$ and $\Omega_l^{\mathrm{r}}$ denote its departure and arrival directions. Using the TX and RX codebook vectors $\mathbf{w}_t^{(b_t)}$ and $\mathbf{w}_r^{(b_r)}$, respectively, the ground-truth power is

$$P_{\mathrm{gt}}(b_t, b_r) = \sum_{l} g_l \left| \mathbf{w}_t^{(b_t)\mathsf{H}} \mathbf{a}_t(\Omega_l^{\mathrm{t}}) \right|^2 \left| \mathbf{w}_r^{(b_r)\mathsf{H}} \mathbf{a}_r(\Omega_l^{\mathrm{r}}) \right|^2. \tag{6}$$

The exhaustive sweep evaluates all beam pairs and selects the pair with the largest $P_{\mathrm{gt}}$, providing an oracle reference.

**Top-$K$ selection and spectral efficiency.** RFWeave ranks all TX/RX beam pairs according to $\widehat{G}(b_t, b_r)$. Let $\mathcal{B}_K$ contain the $K$ highest-ranked pairs. To evaluate beam ranking independently of received power calibration, the predicted scores are used only to rank the candidates. Achieved spectral efficiency is then computed using the ground-truth power of the retained beam pairs,

$$R^{(K)} = \max_{(b_t, b_r) \in \mathcal{B}_K} \log_2 \left( 1 + \frac{P_{\mathrm{gt}}(b_t, b_r)}{P_{\mathrm{n}}} \right). \tag{7}$$

Here, $P_{\mathrm{n}}$ denotes the noise power, computed from $N_0 = -174$ dBm/Hz over a bandwidth of $BW = 100$ MHz and converted to linear units. For Top-1, RFWeave retains only its highest-ranked beam pair. Top-2 and Top-3 retain the two and three highest-ranked candidates, respectively. The best spectral efficiency among the retained pairs is then evaluated using ground-truth beam-pair power. Location-based steering produces a single beam pair and is therefore reported only for Top-1. The exhaustive sweep evaluates all beam pairs and provides the oracle upper bound. Table 6 reports the mean and median spectral efficiency under this protocol.