Supplementary Material for
Generalizable Radio-Frequency Radiance Fields for Spatial Spectrum Synthesis

Appendix Contents

A. Proof of the Interpolation Theory in the RF Domain

We first present the mathematical proof, followed by empirical validation of the interpolation theory in the RF domain. Specifically, we show that the spatial spectrum $\mathbf{SS}(\mathbf{P})$ at any position $\mathbf{P}$ can be approximated as a weighted combination of the spectra from its L-nearest neighbors $\mathcal{N}_L(\mathbf{P})$:

A.1. Mathematical Proof

We present a mathematical proof demonstrating that the spatial spectrum $\mathbf{SS}_t$ at a target transmitter position $\mathbf{P}_t$ can be approximated as a combination of spatial spectra $\{\mathbf{SS}_i\}_{i=1}^L$ corresponding to L neighboring transmitter positions $\{\mathbf{P}_i\}_{i=1}^L$. The proof is structured in three steps: a Taylor series expansion, a convex linear combination, and an error analysis.

Assumptions: The proof presupposes that transmitters and receivers equipped with isotropic antennas are placed in free 3D space, under the following two assumptions:

ASSUMPTION 1: A smooth propagation environment, wherein each transmitter position $\mathbf{P} \in \mathbb{R}^3$ is mapped to a corresponding spatial spectrum $\mathbf{SS} \in \mathbb{R}^{360\times 90}$ through a function $\mathcal{T} : \mathbb{R}^3 \to \mathbb{R}^{360\times 90}$. The mapping $\mathcal{T}$ is assumed to be twice continuously differentiable with respect to the position $\mathbf{P} = (x, y, z)$. This regularity condition ensures that the spatial spectrum $\mathbf{SS} = \mathcal{T}(\mathbf{P})$ and its first- and second-order derivatives vary smoothly across the local region of $\mathbf{P}$, a standard postulate in physics-based analyses of wave propagation [2, 5].

For example, consider the Friis Free Space Propagation Equation [10], analogous to the mapping $\mathcal{T}$, where the received signal power S_{rx} at a receiver position $\mathbf{P}_{rx}$ is given by (note that $S_{rx} \in \mathbb{R}$ is a scalar, whereas our setting involves a full spatial spectrum $\mathbf{SS} \in \mathbb{R}^{360\times 90}$):

$$S_{rx} = S_{tx}\left(\frac{\lambda}{4\pi d}\right)^2, \quad \text{with} \quad d = \|\mathbf{P}_{tx} - \mathbf{P}_{rx}\|$$

where S_{tx} is the transmitted power, λ is the wavelength, and d is the Euclidean distance between the transmitter at $\mathbf{P}_{tx}$ and the receiver at $\mathbf{P}_{rx}$, which is a smooth function of position $\mathbf{P}$ for $d > 0$. Since $S_{rx} \propto d^{-2}$, the received power also varies smoothly with respect to positional changes in $\mathbf{P}$.

While the Friis model yields a single scalar value, the mapping $\mathcal{T}$ generalizes this concept to produce a spatial spectrum $\mathbf{SS} \in \mathbb{R}^{360\times 90}$, representing signal power across all directions around the receiver, with smoothness similarly ensured by continuous variation in free space.

ASSUMPTION 2: We adopt the standard interpolation assumption in $\mathbb{R}^3$: it is possible to identify neighboring positions $\{\mathbf{P}_i\}_{i=1}^L$ such that the target position $\mathbf{P}_t$ lies within their convex hull, i.e.,

$$\sum_{i=1}^{L} w_i \mathbf{P}_i = \mathbf{P}_t$$

where the w_i are barycentric weights satisfying $w_i \geq 0$ and $\sum_{i=1}^L w_i = 1$.

Taylor Series Expansion of the Spatial Spectrum.

Objective: Derive the relationship between the spatial spectra at two proximate transmitter positions, $\mathbf{P}_i$ and $\mathbf{P}_j$, where $\mathbf{P}_j = \mathbf{P}_i + \Delta\mathbf{P}_{ij}$ with $\Delta\mathbf{P}_{ij} = (x_j - x_i,\, y_j - y_i,\, z_j - z_i)$, by applying a Taylor series expansion under ASSUMPTION 1.

Statement:

$$\mathbf{SS}_j = \mathcal{T}(\mathbf{P}_i + \Delta\mathbf{P}_{ij}) = \mathbf{SS}_i + \nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_i)\cdot\Delta\mathbf{P}_{ij} + \frac{1}{2}\Delta\mathbf{P}_{ij}^{\top}\nabla_{\mathbf{P}}^2\mathcal{T}(\mathbf{P}_i)\,\Delta\mathbf{P}_{ij} + \mathcal{O}\left(\|\Delta\mathbf{P}_{ij}\|^3\right)$$

Explanation:

- **Preliminaries**:
  - Spectra $\mathbf{SS}_i$ and $\mathbf{SS}_j$ are defined through a mapping $\mathcal{T} : \mathbb{R}^3 \to \mathbb{R}^{360\times 90}$, where $\mathbf{SS}_i = \mathcal{T}(\mathbf{P}_i)$ and $\mathbf{SS}_j = \mathcal{T}(\mathbf{P}_j)$, for transmitter positions $\mathbf{P}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ and $\mathbf{P}_j = \mathbf{P}_i + \Delta\mathbf{P}_{ij}$, with $\Delta\mathbf{P}_{ij}$ denoting the displacement vector between $\mathbf{P}_i$ and $\mathbf{P}_j$.
  - The magnitude of the displacement $\|\Delta\mathbf{P}_{ij}\|$ is assumed to be sufficiently small, ensuring that geometric perturbations remain minimal.

- **Terms**:
  - $\nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_i) = \left(\frac{\partial \mathcal{T}}{\partial x}, \frac{\partial \mathcal{T}}{\partial y}, \frac{\partial \mathcal{T}}{\partial z}\right) \in \mathbb{R}^{360\times 90\times 3}$: The gradient tensor, which quantifies the first-order sensitivity of the spatial spectrum to changes in the coordinates of $\mathbf{P}_i$.
  - $\nabla^2_{\mathbf{P}}\mathcal{T}(\mathbf{P}_i) \in \mathbb{R}^{360\times 90\times 3\times 3}$: The Hessian tensor, comprising second-order partial derivatives and capturing the curvature of the spatial spectrum with respect to $\mathbf{P}_i$.
  - $\mathcal{O}\left(\|\Delta\mathbf{P}_{ij}\|^3\right) \in \mathbb{R}^{360\times 90}$: The remainder term, which encompasses third- and higher-order derivatives and becomes negligible when $\Delta\mathbf{P}_{ij}$ is sufficiently small.

Interpolating the Target Spectrum via Weighted Combination.

Objective: Estimate the spatial spectrum $\mathbf{SS}_t$ at a target position $\mathbf{P}_t \in \mathbb{R}^3$ by blending the known spectra $\{\mathbf{SS}_i\}_{i=1}^L$ from nearby transmitter positions $\{\mathbf{P}_i\}_{i=1}^L \subset \mathbb{R}^3$.

Statement: Given the Taylor expansion from Step 1 and the barycentric weights (ASSUMPTION 2), the spatial spectrum at the target position $\mathbf{P}_t$ can be approximated as:

$$\mathbf{SS}_t = \sum_{i=1}^{L} w_i \mathbf{SS}_i + \sum_{i=1}^{L} w_i \nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_i)\cdot(\mathbf{P}_t - \mathbf{P}_i) + \mathcal{O}\left(\|\Delta\mathbf{P}\|^2\right)$$

where $\Delta\mathbf{P} = \max_i \|\mathbf{P}_t - \mathbf{P}_i\|$ denotes the maximum distance between the target position $\mathbf{P}_t$ and its neighboring transmitter positions $\{\mathbf{P}_i\}_{i=1}^L$. Given the Lipschitz continuity of $\nabla_{\mathbf{P}}\mathcal{T}$ (which follows from the mapping function $\mathcal{T}$ being twice continuously differentiable), the first-order term is bounded by $\mathcal{O}\left(\|\Delta\mathbf{P}\|^2\right)$, thereby simplifying the expression to:

$$\mathbf{SS}_t = \sum_{i=1}^{L} w_i \mathbf{SS}_i + \mathcal{O}\left(\|\Delta\mathbf{P}\|^2\right)$$

Explanation:

- **Construction**:
  - For each neighbor $\mathbf{P}_i$, we use the Taylor expansion around $\mathbf{P}_i$ from Step 1:

    $$\mathbf{SS}_t = \mathbf{SS}_i + \nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_i)\cdot(\mathbf{P}_t - \mathbf{P}_i) + \mathcal{O}\left(\|\mathbf{P}_t - \mathbf{P}_i\|^2\right)$$

    This provides a local approximation of $\mathbf{SS}_t$ when $\mathbf{P}_t$ is close to $\mathbf{P}_i$.
  - We combine these approximations using barycentric weights $\{w_i\}_{i=1}^L$, where $w_i \geq 0$, $\sum_{i=1}^L w_i = 1$, and $\sum_{i=1}^L w_i\mathbf{P}_i = \mathbf{P}_t$. The weighted estimate becomes:

    $$\mathbf{SS}_t = \sum_{i=1}^{L} w_i \mathbf{SS}_i + \sum_{i=1}^{L} w_i \nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_i)\cdot(\mathbf{P}_t - \mathbf{P}_i) + \sum_{i=1}^{L} w_i \mathcal{O}\left(\|\mathbf{P}_t - \mathbf{P}_i\|^2\right)$$

- **Simplification**:
  - *Bounding the First-Order Term*: The barycentric property ensures:

    $$\sum_{i=1}^{L} w_i(\mathbf{P}_t - \mathbf{P}_i) = \sum_{i=1}^{L} w_i\mathbf{P}_t - \sum_{i=1}^{L} w_i\mathbf{P}_i = \mathbf{P}_t - \mathbf{P}_t = \mathbf{0}$$

    However, the gradients $\nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_i)$ vary across positions $\mathbf{P}_i$. Since $\mathcal{T}$ is twice differentiable, its gradient $\nabla_{\mathbf{P}}\mathcal{T}$ is Lipschitz continuous. Thus, we have:

    $$\nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_i) = \nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_t) + \mathbf{E}_i, \quad \text{with} \quad \|\mathbf{E}_i\| \leq K_1\|\mathbf{P}_t - \mathbf{P}_i\|$$

    where K_1 is the Lipschitz constant of $\nabla_{\mathbf{P}}\mathcal{T}$, quantifying the maximum rate of change of the gradient, with $K_1 = 2K$ and $K = \frac{1}{2}\sup_{\mathbf{P}\in\text{conv}\{\mathbf{P}_i\}}\left\|\nabla^2_{\mathbf{P}}\mathcal{T}(\mathbf{P})\right\|$ as defined in Step 3. Substituting into the first-order term, we obtain:

    $$\sum_{i=1}^{L} w_i \nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_i)\cdot(\mathbf{P}_t - \mathbf{P}_i) = \sum_{i=1}^{L} w_i\left[\nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_t) + \mathbf{E}_i\right]\cdot(\mathbf{P}_t - \mathbf{P}_i)$$

This splits as:

$$\nabla_{\mathbf{P}}\mathcal{T}(\mathbf{P}_t)\cdot\sum_{i=1}^{L}w_i\left(\mathbf{P}_t-\mathbf{P}_i\right)+\sum_{i=1}^{L}w_i\mathbf{E}_i\cdot\left(\mathbf{P}_t-\mathbf{P}_i\right)=0+\sum_{i=1}^{L}w_i\mathbf{E}_i\cdot\left(\mathbf{P}_t-\mathbf{P}_i\right)$$

Bounding the error term:

$$\left\|\sum_{i=1}^{L}w_i\mathbf{E}_i\cdot\left(\mathbf{P}_t-\mathbf{P}_i\right)\right\|\leq\sum_{i=1}^{L}w_i\left\|\mathbf{E}_i\right\|\left\|\mathbf{P}_t-\mathbf{P}_i\right\|\leq K_1\sum_{i=1}^{L}w_i\left\|\mathbf{P}_t-\mathbf{P}_i\right\|^2\leq K_1\left\|\Delta\mathbf{P}\right\|^2$$

where $\sum_{i=1}^{L}w_i=1$. Therefore, the first-order term is bounded by $\mathcal{O}\left(\left\|\Delta\mathbf{P}\right\|^2\right)$.

- *Residual Error Term*: The higher-order terms are:

$$\sum_{i=1}^{L}w_i\mathcal{O}\left(\left\|\mathbf{P}_t-\mathbf{P}_i\right\|^2\right)$$

Since $\left\|\mathbf{P}_t-\mathbf{P}_i\right\|\leq\left\|\Delta\mathbf{P}\right\|$ and $\sum_{i=1}^{L}w_i=1$, we have:

$$\sum_{i=1}^{L}w_i\mathcal{O}\left(\left\|\mathbf{P}_t-\mathbf{P}_i\right\|^2\right)\leq\mathcal{O}\left(\left\|\Delta\mathbf{P}\right\|^2\right)$$

- *Final Result*: Combining all terms:

$$\mathbf{SS}_t=\sum_{i=1}^{L}w_i\mathbf{SS}_i+\mathcal{O}\left(\left\|\Delta\mathbf{P}\right\|^2\right)+\mathcal{O}\left(\left\|\Delta\mathbf{P}\right\|^2\right)=\sum_{i=1}^{L}w_i\mathbf{SS}_i+\mathcal{O}\left(\left\|\Delta\mathbf{P}\right\|^2\right)$$

The error is quadratic in $\left\|\Delta\mathbf{P}\right\|$, meaning it decreases proportionally to the square of the maximum distance between the target position and its neighboring transmitters.

Quantifying the Interpolation Error.

Objective: Derive an upper bound on the interpolation error to assess the approximation accuracy.

Statement: The interpolation error $\epsilon=\left\|\mathbf{SS}_t-\sum_{i=1}^{L}w_i\mathbf{SS}_i\right\|$ satisfies the bound:

$$\epsilon\leq K\cdot\max_i\left\|\mathbf{P}_t-\mathbf{P}_i\right\|^2,\quad\text{with}\quad K=\frac{1}{2}\sup_{\mathbf{P}\in\text{conv}\{\mathbf{P}_i\}}\left\|\nabla_{\mathbf{P}}^2\mathcal{T}\left(\mathbf{P}\right)\right\|$$

where K is a constant that depends on the second-order derivatives of the function $\mathcal{T}$, and encapsulates the maximum curvature of $\mathcal{T}$ over the convex hull of the neighboring positions $\{\mathbf{P}_i\}_{i=1}^{L}$.

Explanation:

- **Derivation**:
  - From Step 2, the interpolation error ϵ is expressed as:

$$\epsilon=\left\|\mathbf{SS}_t-\sum_{i=1}^{L}w_i\mathbf{SS}_i\right\|=\left\|\mathcal{O}\left(\left\|\Delta\mathbf{P}\right\|^2\right)\right\|$$

  - $\mathcal{O}\left(\left\|\Delta\mathbf{P}\right\|^2\right)$ is bounded by the second derivatives of $\mathcal{T}$. By Taylor's theorem, for a twice continuously differentiable function $\mathcal{T}$, there exists a constant $K>0$ such that:

$$\left\|\mathcal{T}\left(\mathbf{P}_t\right)-\mathcal{T}\left(\mathbf{P}_i\right)-\nabla_{\mathbf{P}}\mathcal{T}\left(\mathbf{P}_i\right)\cdot\left(\mathbf{P}_t-\mathbf{P}_i\right)\right\|\leq K\left\|\mathbf{P}_t-\mathbf{P}_i\right\|^2$$

where $K=\frac{1}{2}\sup_{\mathbf{P}\in\text{conv}\{\mathbf{P}_i\}}\left\|\nabla_{\mathbf{P}}^2\mathcal{T}\left(\mathbf{P}\right)\right\|$ bounds the acceleration of the spatial spectrum's variation across the interpolation region. Smoother environments, those with gentler second-order changes, yield smaller K, tightening the error bound $\epsilon\leq K\cdot\left\|\Delta\mathbf{P}\right\|^2$. For example, in free-space propagation, K is negligible, whereas near obstacles, where signal strength fluctuates rapidly, K grows significantly.

* $\nabla^2_{\mathbf{P}}\mathcal{T}(\mathbf{P})$ is the *Hessian matrix* of $\mathcal{T}$ at $\mathbf{P}$, quantifying how sensitively the spectrum changes in response to second-order variations in the transmitter coordinates.
* $\left\|\nabla^2_{\mathbf{P}}\mathcal{T}(\mathbf{P})\right\|$ denotes the *operator norm* of the Hessian, a scalar measure of its "magnitude," representing the maximum rate of curvature of $\mathcal{T}$ in any direction.
* $\sup_{\mathbf{P}\in\text{conv}\{\mathbf{P}_i\}}$ takes the *supremum* (least upper bound) of the Hessian norm over all points in the convex hull of the neighboring positions $\{\mathbf{P}_i\}$. This represents the worst-case curvature of $\mathcal{T}$ within the local interpolation region.

– Summing over all neighbors' spatial spectra with barycentric weights w_i, and noting that $\sum_{i=1}^{L} w_i = 1$, the total error satisfies:

$$\epsilon \leq \sum_{i=1}^{L} w_i \cdot K \left\|\mathbf{P}_t - \mathbf{P}_i\right\|^2 \leq K \cdot \max_i \left\|\mathbf{P}_t - \mathbf{P}_i\right\|^2$$

- **Intuition**:
  – **Interpretability of** K: The constant K directly quantifies the worst-case curvature of $\mathcal{T}$ over the interpolation region, as governed by the Hessian $\nabla^2_{\mathbf{P}}\mathcal{T}$. In free-space propagation (low curvature, small K), the bound tightens significantly, whereas near obstacles or multipath-rich environments (high curvature, large K), the error grows predictably. This ties the theoretical guarantee to physically observable phenomena, ensuring the result is not merely abstract but grounded in wave physics.
  – **Scope of the Error Bound**: The derived bound applies specifically to the linear convex combination of neighboring spectra, as defined by barycentric weights. While this provides a foundational guarantee for basic linear interpolation, the *GRaF* architecture extends beyond this linear regime. By integrating a neural network, *GRaF* learns nonlinear corrections to the weighted spectral average, capturing higher-order correlations and environmental discontinuities (e.g., diffraction, reflection, scattering) that the linear model cannot represent. Thus, the proven bound serves as a baseline for the worst-case error in linear neighbor spatial spectrum interpolation.

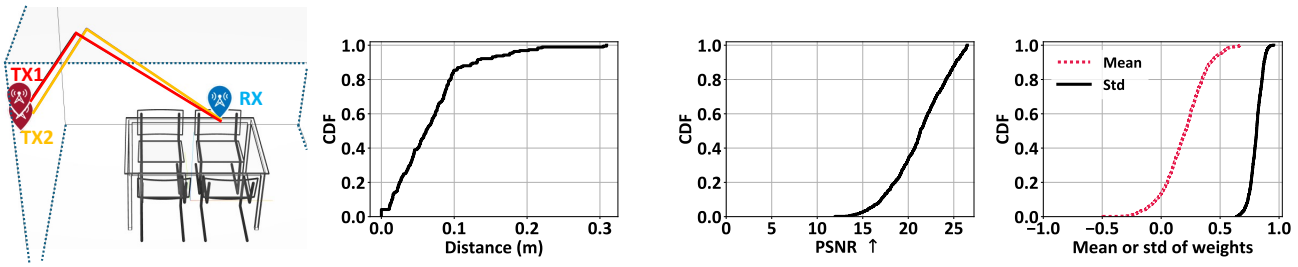
A.2. Empirical Validation

The rationale is based on two observations highlighted in this subsection. First, for a given target transmitter, neighboring transmitters typically traverse similar propagation paths, *i.e.*, they pass through many common voxels within the scene. Second, the spectrum for the target transmitter can be perceived as an interpolation of the spectra from its neighbors. Consequently, leveraging the spectra from these neighbors facilitates the spectrum generation for the target transmitter, thereby improving the generalization ability as long as neighbor data is available. We conduct two empirical experiments to demonstrate these two observations.

Observation #1. In a conference room, as depicted in Figure 1a, we position the RX at a fixed location and place TX1 and TX2 at two closely situated positions. We utilize the MATLAB ray tracing simulation [8] to conduct RF ray tracing analysis. This software calculates all the propagation paths between the transmitters and the receiver. For brevity, we present one path from either TX1 or TX2 to the RX in Figure 1a. It is evident that the starting and reflection positions of the two paths are very close, and they both terminate at the RX.

For a more general case, we generate 500 pairs of closely positioned TX1 and TX2, execute the RF Insite ray tracing algorithm for each pair, and then compare the distances between the reflection points of their corresponding paths. Figure 1a shows that approximately 80% of the distances between reflection points are less than 0.1 m. Given their close starting positions, proximate reflection points, and identical end positions, these factors collectively indicate a similar propagation path. Moreover, assuming the voxel size equals the wavelength, such as 0.13 m for 2.4 GHz WiFi, the path differences are typically less than the voxel size. Therefore, the propagation paths for closely positioned transmitters typically pass through many common voxels.

Observation #2. Given that closely positioned transmitters pass through many common voxels, their spectra may exhibit some correlations. To verify this, we conduct an experiment using RFID dataset (details in Section Experiments). For each transmitter and its corresponding spectrum, we identify the 6 closest positions of other transmitters as its neighbors. We then assign a weight matrix to each neighbor; since the spectrum size is $(360, 90)$, each weight matrix is also $(360, 90)$, with every pixel of the spectrum assigned a specific weight. Using each neighbor's spectrum and its assigned weight matrix, we perform a weighted summation of all neighbors' spectra to predict the target transmitter's spectrum. We employ the MSE loss to train the 6 weight matrices over 200 iterations for each target transmitter. Once trained, we compare the generated spatial spectrum with the ground truth spectrum.



(a) Propagation path (left) and reflection distance (right)

(b) PSNR comparison (left) and interpolation statistics (right)

Figure 1. (Left) Propagation path differences and reflection distance between two transmitters (TXs) and a receiver (RX). (Right) PSNR when using neighbors' spectra and statistics of the interpolation matrix.

Figure 1b illustrates that an average PSNR of 20.5 dB with a standard deviation of 4.1 dB is achieved for the interpolated spectra. To understand what the value of 20.5 dB represents, refer to the second row, first column of Figure 4, which displays a PSNR of 19.5 dB between two spatial spectra. Visually, the similarity between them is apparent. Thus, an average PSNR of 20.5 dB suggests that the neighbors' spectra are indeed closely related to the target transmitter's spectrum. Hence, the target transmitter's spectrum can be viewed as an interpolation of the spectra of its neighbors. Explaining this phenomenon from a broader perspective, novel view synthesis in the context of visible light can be achieved by interpolating between images captured from different camera poses and views [1]. Similarly, RF signals may also exhibit such properties. Because visible light and RF signals are both forms of electromagnetic waves, they might share underlying interpolation characteristics, despite their differences in propagation behaviors.

You might wonder whether this interpolation method could be directly applied to synthesize spectra for a target transmitter. The answer is NO. First, although the generated spectrum shows promise, it is not sufficiently accurate. Second, the weight matrices are highly dependent on the transmitter positions. A set of weight matrices from one transmitter cannot be applied to another transmitter. We normalize all learned weight matrices for all transmitters and present their mean and standard deviation in Figure 1b. It can be observed that the values range from -1.0 to 1.0, covering all possible values. Thus, the values of the weight matrix are highly dynamic, and a single set of weight matrices cannot be used for other transmitters.

B. Derivation of the Received Signal in Neural Ray Tracing

Objective.

The objective is to derive the expression for the received signal y_r in the neural ray tracing algorithm:

$$y_r = \sum_{s=1}^{S} \left(\prod_{j=1}^{s-1} a\left(\mathbf{x}_j, \alpha, \beta\right) \right) s\left(\mathbf{x}_s, \alpha, \beta\right) \cdot \frac{\lambda}{4\pi d_s} e^{-j\frac{2\pi f d_s}{c}}. \tag{1}$$

It models the RF signal received at the receiver by aggregating contributions from all sampled points along the ray, considering both attenuation and phase shifts introduced during propagation.

Maxwell's Equations and the Wave Equation.

We begin with Maxwell's equations, which govern electromagnetic wave propagation. In free space (*i.e.*, no charges or currents), the electric field $\mathbf{E}$ satisfies the wave equation. Taking the curl of Faraday's law and applying the constitutive relations, we derive the wave equation:

$$\nabla^2 \mathbf{E} + k^2 \mathbf{E} = 0,$$

where:

- ∇^2 is the Laplacian operator, representing spatial changes in the field,
- $k = \frac{2\pi f}{c} = \frac{2\pi}{\lambda}$ is the wave number,
- f is the signal frequency,
- c is the speed of light,
- λ is the wavelength.

This is the Helmholtz equation, which describes how electromagnetic waves propagate through space. For clarity, we simplify the analysis to a scalar field, representing a single component of $\mathbf{E}$.

Green's Function: Modeling a Point Source.

To solve the Helmholtz equation for a point source (such as a radiating point along a ray), we employ the Green's function. For a source located at position $\mathbf{x}'$, the resulting field at position $\mathbf{x}$ is:

$$G(\mathbf{x}, \mathbf{x}') = \frac{e^{-jk\|\mathbf{x}-\mathbf{x}'\|}}{4\pi \|\mathbf{x}-\mathbf{x}'\|},$$

where:

- **Physical Meaning:** Represents a spherical wave emanating from $\mathbf{x}'$.
- **Terms:**
  - $e^{-jk\|\mathbf{x}-\mathbf{x}'\|}$: Phase shift induced by traveling the distance $\|\mathbf{x}-\mathbf{x}'\|$,
  - $\frac{1}{4\pi\|\mathbf{x}-\mathbf{x}'\|}$: Amplitude decay due to spherical spreading,
  - $k = \frac{2\pi}{\lambda}$: Wave number, relating phase to wavelength.

This serves as the fundamental block for modeling wave propagation from a single point source.

Ray Tracing: Discretizing the Path.

In ray tracing, the signal path is modeled as a series of discrete points $\mathbf{x}_s$, where $s = 1, 2, \ldots, S$. Each point acts as a secondary source, radiating its own contribution to the received signal. The field at the receiver $\mathbf{P}_{\text{rx}}$ from a point $\mathbf{x}_s$ is expressed as:

$$E_s(\mathbf{P}_{\text{rx}}) = s(\mathbf{x}_s, \alpha, \beta) \cdot \frac{e^{-jkd_s}}{4\pi d_s},$$

where:

- $s(\mathbf{x}_s, \alpha, \beta)$: The complex signal (amplitude and phase) emitted or scattered at $\mathbf{x}_s$, dependent on position and parameters α, β,
- $d_s = \|\mathbf{x}_s - \mathbf{P}_{\text{rx}}\|$: The Euclidean distance from $\mathbf{x}_s$ to the receiver,
- $\frac{e^{-jkd_s}}{4\pi d_s}$: The Green's function applied over the path distance, accounting for spherical wave spreading and phase shift.

The signal at $\mathbf{x}_s$ radiates as a spherical wave, and this formulation computes its contribution to the total field strength at the receiver.

Adding Path Attenuation.

As the signal propagates from the transmitter to $\mathbf{x}_s$, it undergoes interactions such as reflections and absorptions at earlier points $\mathbf{x}_j$, where $j = 1, 2, \ldots, s-1$. We model this cumulative effect as:

$$\prod_{j=1}^{s-1} a(\mathbf{x}_j, \alpha, \beta),$$

- **Physical Meaning:** Each $a(\mathbf{x}_j, \alpha, \beta)$ is a complex factor that represents attenuation (magnitude less than 1) and phase shift induced by the material or obstacle at $\mathbf{x}_j$.
- **Multiplicative Nature:** These factors multiply sequentially because each segment modifies the signal as it progresses through the path.

Thus, the contribution from the sampled point $\mathbf{x}_s$ to the received signal becomes:

$$\left(\prod_{j=1}^{s-1} a(\mathbf{x}_j, \alpha, \beta)\right) s(\mathbf{x}_s, \alpha, \beta) \cdot \frac{e^{-jkd_s}}{4\pi d_s}.$$

This expression incorporates both the path attenuation effects accumulated from previous points and the radiated field from $\mathbf{x}_s$, considering distance-based decay and phase shift as modeled by the Green's function.

Summing All Contributions.

The received signal y_r is obtained by summing the contributions from all S sampled points along the ray path:

$$y_r = \sum_{s=1}^{S} \left(\prod_{j=1}^{s-1} a\left(\mathbf{x}_j, \alpha, \beta\right) \right) s\left(\mathbf{x}_s, \alpha, \beta\right) \cdot \frac{e^{-jkd_s}}{4\pi d_s}.$$

- **Why Sum?** In wave propagation, fields from multiple sources superpose linearly, assuming linearity of the medium.
- **Adjustment:** Substitute $k = \frac{2\pi f}{c}$, and recognize that $\frac{1}{4\pi d_s}$ can be scaled by the wavelength λ for dimensional consistency:

$$y_r = \sum_{s=1}^{S} \left(\prod_{j=1}^{s-1} a\left(\mathbf{x}_j, \alpha, \beta\right) \right) s\left(\mathbf{x}_s, \alpha, \beta\right) \cdot \frac{\lambda}{4\pi d_s} e^{-j\frac{2\pi f d_s}{c}}.$$

This formulation links electromagnetic theory with the neural ray tracing model, enabling accurate RF signal prediction by accounting for path attenuation, phase shifts, and spherical spreading.

C. Proof of Loss Function

We prove that the optimization problem for maximizing the log-likelihood of the spatial spectrum:

$$\Theta^* = \arg\max_{\Theta} \log p\left(\mathbf{SS}_\Theta \mid \mathcal{N}_L, \mathbf{P}\right),$$

where:

$$\log p\left(\mathbf{SS}_\Theta \mid \mathcal{N}_L, \mathbf{P}\right) = \log \int \prod_{r=1}^{Q} p\left(\mathbf{SS}\left(r\right) \mid \mathbf{Z}, \mathbf{P}\right) p\left(\mathbf{Z} \mid \mathcal{N}_L\right) \, d\mathbf{Z},$$

can be reformulated as minimizing the ℓ_2 reconstruction loss:

$$\Theta^* = \arg\min_{\Theta} \sum_{r=1}^{Q} \left\| \mathbf{SS}\left(r\right) - \hat{\mathbf{SS}}_\Theta\left(r\right) \right\|^2,$$

given that $\mathbf{Z}$ is deterministically computed by the latent RF radiance field as $\mathbf{Z} = \mathcal{T}_\Psi\left(\mathcal{N}_L, \mathbf{P}\right)$.

Simplifying the Integral with Deterministic Z.

The likelihood involves an integral over the latent variable $\mathbf{Z}$:

$$p\left(\mathbf{SS}_\Theta \mid \mathcal{N}_L, \mathbf{P}\right) = \int \prod_{r=1}^{Q} p\left(\mathbf{SS}\left(r\right) \mid \mathbf{Z}, \mathbf{P}\right) p\left(\mathbf{Z} \mid \mathcal{N}_L\right) \, d\mathbf{Z}.$$

Since $\mathbf{Z}$ is deterministically computed by the function $\mathcal{T}_\Psi$, *i.e.*, $\mathbf{Z} = \mathcal{T}_\Psi\left(\mathcal{N}_L, \mathbf{P}\right)$, the conditional distribution $p\left(\mathbf{Z} \mid \mathcal{N}_L\right)$ is a Dirac delta function:

$$p\left(\mathbf{Z} \mid \mathcal{N}_L\right) = \delta\left(\mathbf{Z} - \mathcal{T}_\Psi\left(\mathcal{N}_L, \mathbf{P}\right)\right).$$

Substituting this into the integral:

$$p\left(\mathbf{SS}_\Theta \mid \mathcal{N}_L, \mathbf{P}\right) = \int \prod_{r=1}^{Q} p\left(\mathbf{SS}\left(r\right) \mid \mathbf{Z}, \mathbf{P}\right) \delta\left(\mathbf{Z} - \mathcal{T}_\Psi\left(\mathcal{N}_L, \mathbf{P}\right)\right) \, d\mathbf{Z}.$$

Using the property of the Dirac delta function:

$$\int f\left(\mathbf{Z}\right) \delta\left(\mathbf{Z} - \mathbf{Z}_0\right) d\mathbf{Z} = f\left(\mathbf{Z}_0\right),$$

the integral evaluates to the integrand at $\mathbf{Z} = \mathcal{T}_\Psi(\mathcal{N}_L, \mathbf{P})$:

$$p(\mathbf{SS}_\Theta \mid \mathcal{N}_L, \mathbf{P}) = \prod_{r=1}^{Q} p(\mathbf{SS}(r) \mid \mathcal{T}_\Psi(\mathcal{N}_L, \mathbf{P}), \mathbf{P}).$$

Taking the logarithm:

$$\log p(\mathbf{SS}_\Theta \mid \mathcal{N}_L, \mathbf{P}) = \log \prod_{r=1}^{Q} p(\mathbf{SS}(r) \mid \mathcal{T}_\Psi(\mathcal{N}_L, \mathbf{P}), \mathbf{P}) = \sum_{r=1}^{Q} \log p(\mathbf{SS}(r) \mid \mathcal{T}_\Psi(\mathcal{N}_L, \mathbf{P}), \mathbf{P}).$$

Thus, the optimization problem becomes:

$$\Theta^* = \arg\max_\Theta \sum_{r=1}^{Q} \log p(\mathbf{SS}(r) \mid \mathcal{T}_\Psi(\mathcal{N}_L, \mathbf{P}), \mathbf{P}).$$

Modeling the Conditional Likelihood.

To proceed, we specify the conditional likelihood $p(\mathbf{SS}(r) \mid \mathbf{Z}, \mathbf{P})$. In supervised learning tasks involving continuous outputs, such as reconstructing spatial spectra, it is standard to assume the observations follow a Gaussian distribution centered at the model's prediction. Let $\hat{\mathbf{SS}}_\Theta(r)$ denote the predicted spatial spectrum for ray r, parameterized by Θ. We assume:

$$p(\mathbf{SS}(r) \mid \mathbf{Z}, \mathbf{P}) = \mathcal{N}\left(\mathbf{SS}(r) \mid \hat{\mathbf{SS}}_\Theta(r), \sigma^2\right),$$

where $\mathcal{N}$ denotes a Gaussian distribution with mean $\hat{\mathbf{SS}}_\Theta(r)$ and variance σ^2, and $\mathbf{Z} = \mathcal{T}_\Psi(\mathcal{N}_L, \mathbf{P})$. The Gaussian probability density function is given by:

$$p(\mathbf{SS}(r) \mid \mathbf{Z}, \mathbf{P}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \left\|\mathbf{SS}(r) - \hat{\mathbf{SS}}_\Theta(r)\right\|^2\right).$$

Taking the logarithm:

$$\log p(\mathbf{SS}(r) \mid \mathbf{Z}, \mathbf{P}) = \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{1}{2\sigma^2} \left\|\mathbf{SS}(r) - \hat{\mathbf{SS}}_\Theta(r)\right\|^2.$$

Thus we have:

$$\log p(\mathbf{SS}(r) \mid \mathbf{Z}, \mathbf{P}) = -\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \left\|\mathbf{SS}(r) - \hat{\mathbf{SS}}_\Theta(r)\right\|^2.$$

Summing over all Q rays:

$$\sum_{r=1}^{Q} \log p(\mathbf{SS}(r) \mid \mathcal{T}_\Psi(\mathcal{N}_L, \mathbf{P}), \mathbf{P}) = \sum_{r=1}^{Q} \left[-\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \left\|\mathbf{SS}(r) - \hat{\mathbf{SS}}_\Theta(r)\right\|^2\right].$$

This can be written as:

$$-\frac{Q}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{r=1}^{Q} \left\|\mathbf{SS}(r) - \hat{\mathbf{SS}}_\Theta(r)\right\|^2.$$

Equivalence to ℓ_2 Loss Minimization.

To maximize the log-likelihood:

$$\Theta^* = \arg\max_\Theta \left[-\frac{Q}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{r=1}^{Q} \left\|\mathbf{SS}(r) - \hat{\mathbf{SS}}_\Theta(r)\right\|^2\right].$$

The first term, $-\frac{Q}{2}\log\left(2\pi\sigma^2\right)$, is constant with respect to Θ, as σ^2 is fixed. Thus, maximizing the log-likelihood is equivalent to minimizing the negative of the second term:

$$\Theta^* = \arg\min_\Theta \frac{1}{2\sigma^2} \sum_{r=1}^{Q} \left\|\mathbf{SS}(r) - \hat{\mathbf{SS}}_\Theta(r)\right\|^2.$$

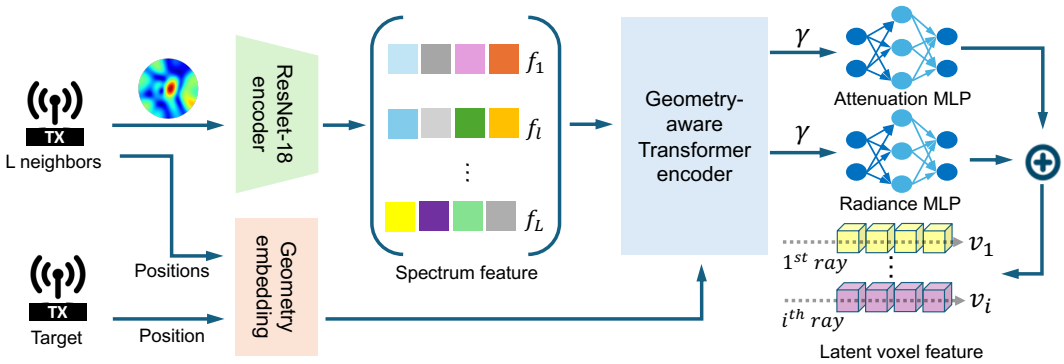


Figure 2. Architecture of geometry-aware RF scene representation module, consisting of a ResNet feature extractor, geometry embedding, a Transformer encoder, and two MLPs, where a circle with a plus symbol signifies concatenation.

Since $\frac{1}{2\sigma^2}$ is a positive constant, it does not affect the optimization. Therefore, the optimization simplifies to:

$$\Theta^* = \arg\min_{\Theta} \sum_{r=1}^{Q} \left\| \mathbf{SS}\left(r\right) - \hat{\mathbf{SS}}_{\Theta}\left(r\right) \right\|^2 .$$

This loss encourages *GRaF* to accurately reconstruct the spatial spectrum while learning contextual latent features from neighboring transmitters' spatial spectra that generalize across diverse scenes.

D. Implementation Details

Training. *GRaF* is trained end-to-end using the L2 loss function, computed per ray as the ray tracing algorithm processes each ray individually, to minimize the difference between the predicted and ground truth signal power. Training is optimized with the Adam optimizer [6], starting with an initial learning rate of 5.0×10^{-4}, which decays exponentially. *GRaF* is trained for 300,000 steps over approximately 15 hours, with $5,120$ rays sampled from the spatial spectrum at each step.

Neighbor Selection Strategy. We select neighbors based on Euclidean distance between transmitter positions, which is directly aligned with our interpolation theory: the approximation error is bounded by $K\,\Delta P^2$, where ΔP is the maximum distance to the selected neighbors and K characterizes the environment curvature. This implies that the spatial proximity of neighbors (quantified by ΔP) and the environment curvature K primarily determine synthesis quality.

To ensure robustness across both dense and sparse regions, we randomly sample $L \in [3, 10]$ for each target transmitter. This encourages the model to learn stable geometric relationships under diverse neighborhood configurations, rather than overfitting to a fixed neighbor count. During testing, neighbors are selected from the training split of the target scene to avoid information leakage. Appendix §E.3 analyzes their impact on synthesis quality.

Because KNN, KNN-DL, and *GRaF* rely exclusively on neighbors from unseen scenes at test time, we fine-tune *NeRF*^{2} on the same data to ensure a fair comparison, as *NeRF*^{2} otherwise requires scene-specific training.

Positional encoding. For the 3D coordinates of transmitter positions, neighbors' positions, and ray positions and directions, we transform these low-dimensional coordinates into high-dimensional representations using the Position Embedding function [11], with 10 frequency bands in all experiments. This expands the 3D coordinates from 3 to $3 + 3 \times 10 \times 2 = 63$ dimensions. In cross-scene experiments, where each transmitter position includes 3D coordinates and a scene index, the 4D coordinates are expanded to $4 + 4 \times 10 \times 2 = 84$ dimensions. This transformation allows the network to distinguish identical coordinates across different scenes.

Voxel Size. The voxel size is consistently set to the wireless signal's wavelength, aligning with the physics of RF propagation. Since phenomena like reflection and diffraction occur at the wavelength scale, this choice ensures that the model captures fine-grained interactions. This design is validated by prior work [7], which solves Maxwell's equations using a finite-volume discretization matched to the electromagnetic wavelength [9].

Radiance Field. Figure 2 illustrates the architecture of our RF radiance field. It comprises a ResNet-18 feature encoder [4], geometry embedding, a geometry-aware Transformer encoder with a cross-attention mechanism [3], and two MLPs.

To extract compact spectral features from the neighboring spectra, we utilize a ResNet-18 encoder with filter settings of 64, 128, 256, and 512 [4]. Both the RF scene representation and the neural ray tracing algorithm are implemented with a Transformer-based architecture. A typical Transformer consists of multiple stacked blocks; each block contains an attention

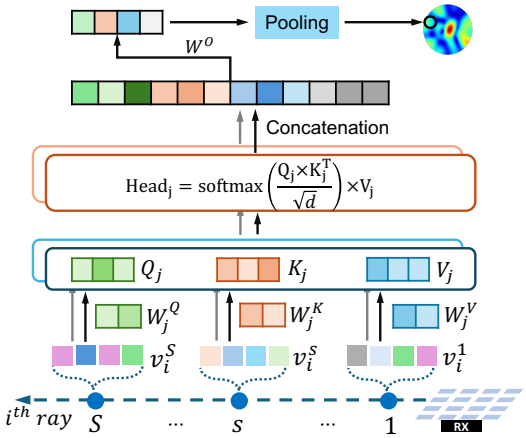


Figure 3. Architecture of the neural-driven ray tracing algorithm, where S voxels are sampled along the i-th ray (RX: receiver).

layer, followed by a residual connection from the input to the post-attention layer, and then layer normalization (LayerNorm). Subsequently, a Feed-Forward Network (FFN) with Rectified Linear Unit (ReLU) activation is applied. This is followed by another residual connection from the output of the first LayerNorm to the output of the ReLU, and the sequence concludes with a second LayerNorm.

The Transformers in both the RF scene representation and the ray tracing algorithm consist of two stacked blocks, each with a hidden dimension of 16, matching the voxel latent feature dimension d. However, there are two key differences between these two Transformers. First, the attention mechanisms differ. The scene representation module incorporates a single-headed cross-attention layer, while the ray tracing Transformer uses a multi-headed self-attention layer. Second, the handling of token sequential information varies. In the scene representation module, the sequential order of spectrum features is not critical, as the focus lies in the differences between pairs of features. Therefore, the index of the spectrum features is not considered. On the other hand, in the ray tracing algorithm, the sequential information of voxels along a ray is essential, as it directly impacts the calculation of attention weights. Therefore, the index of each voxel along the ray is incorporated.

Ray Tracing. Figure 3 illustrates the proposed neural-driven ray tracing algorithm. It utilizes a standard transformer encoder architecture, but here we focus on the core attention mechanism. Along the ray, S voxels are sampled, and their latent features $\{\mathbf{v}^s\}_{s=1,\ldots,S}$ are extracted, with each $\mathbf{v}^s$ treated as a token. The attention scores are computed using the $\mathrm{softmax}$ function applied to the scaled product of the Query matrix Q and Key matrix K, where d is the voxel feature dimension. The scores weight the Value matrix V, resulting in an aggregated latent voxel feature across all S voxels:

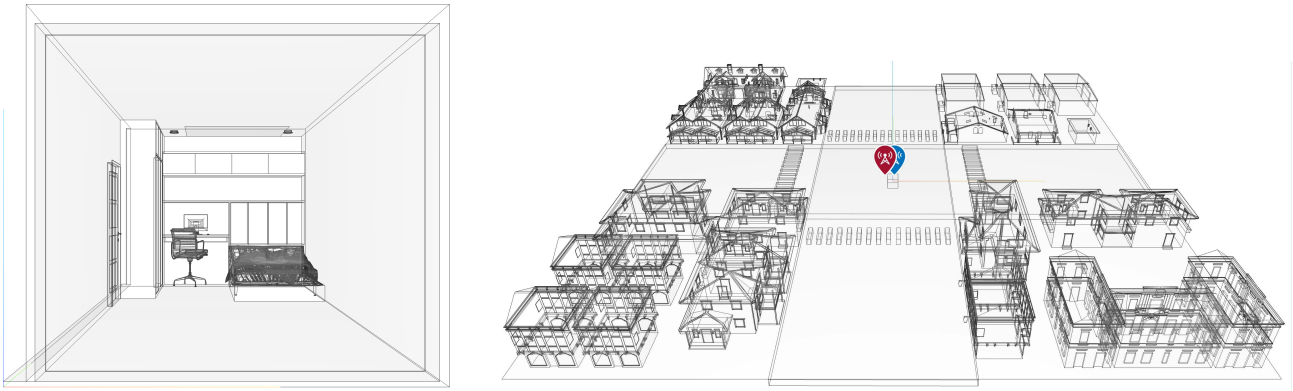
$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}$$

Each voxel feature implicitly encodes the real and imaginary parts of the signal and attenuation, allowing attention scores to capture their complex relationships. Among these, all possible pairwise combinations total $\binom{4+2-1}{2} = 10$, leading to the use of ten attention heads in the transformer to effectively model these dynamics. Each head is equipped with distinct learnable projection matrices $\mathbf{W}_Q^j$, $\mathbf{W}_K^j$, and $\mathbf{W}_V^j$, where j indexes the head from 1 to 10. The output of each head, Head_j, is concatenated into a new feature, which is then multiplied by another learnable projection matrix $\mathbf{W}_o$ to produce the predicted feature for each voxel along the current i-th ray. Finally, mean pooling is performed over all voxels along the ray, and the pooled feature vector is mapped to the real and imaginary parts of the received signal. The amplitude of the resulting signal is calculated as the predicted signal power for the ray.

E. Experimental Results

E.1. Why Simulation Data

GRaF is designed to achieve spatial generalization, enabling adaptation to changes in scene layouts, object placements, and environmental configurations without retraining for each new scene. Simulation offers the precise control necessary for evaluating this capability. Using tools such as Blender to manipulate high-fidelity CAD models, we can systematically modify scene geometry (for example, adding or removing chairs and tables as in ConferenceV1–V3 and OfficeV1–V3) while eliminating confounding factors such as human movement, electromagnetic interference, hardware noise, or temporal drift.



(a) Bedroom layout (b) Outdoor layout

Figure 4. Visualization of two extra layouts for evaluation: a Bedroom scene and an Outdoor scene.

This level of control is essential for rigorously isolating spatial generalization, as demonstrated in our cross-scene experiments where models trained on one layout version (*e.g.*, ConferenceV1) are evaluated on unseen variants (*e.g.*, ConferenceV2/V3) containing structural changes.

Real-world data collection introduces many uncontrolled variables that evolve over time, including changes in occupancy, temperature, humidity, and multipath conditions. Such variability is better suited for studying temporal adaptation or dynamic-scene modeling frameworks (for example, deformable or time-varying NeRFs), rather than spatial generalization.

In contrast, MATLAB-based ray tracing provides high-quality, noise-free supervision for evaluating spectrum synthesis. These simulations incorporate accurate CAD geometry and material properties to model RF propagation phenomena, including reflection, diffraction, scattering, and path loss. The CAD model effectively serves as a physically grounded proxy for the real environment, ensuring that the generated spectra remain geometrically aligned and consistent with Maxwellian propagation behaviors. This controlled and high-fidelity setting is crucial for validating theoretical contributions such as our Spectrum Interpolation Theorem (Theorem 1) and for enabling fair comparison to baseline methods.

To complement simulations and validate real-world applicability, we also evaluate on the RFID dataset, which contains 6,123 real measurements collected at 915 MHz using a 4×4 antenna array. *GRaF* achieves state-of-the-art performance on this dataset, demonstrating compatibility with hardware-collected signals.

Trade-off Between Simulation Quality and Computational Overhead. MATLAB ray tracing provides highly accurate spectra by leveraging detailed CAD models and physics-based rendering. However, this accuracy comes with substantial overhead. Constructing a realistic CAD model requires specialized scanning systems (for example, a $\sim$ \$75K LiDAR scanner) along with significant expert effort to refine geometry and assign accurate material properties. Simulation is also computationally expensive: generating a single spectrum requires an average of 59.5 seconds, resulting in 95.5 hours to generate 5,782 spectra for the bedroom and outdoor layouts (Appendix E.2). This expense creates scalability challenges when building large datasets or iterating over multiple scene variants.

In contrast, *GRaF* operates without any CAD models or pre-existing scene geometry. It requires only transmitter positions and collected spatial spectra, both of which can be obtained using commodity wireless hardware. For example, on the RFID dataset, *GRaF* trains in roughly 15 hours on a single GPU and achieves competitive synthesis quality (21.94 PSNR and 0.766 SSIM in Table 1) alongside strong cross-scene generalization. These characteristics make *GRaF* practical for real deployments such as WiFi-based localization, RFID sensing, and network planning, where rapid adaptation to new environments is essential. By removing the need for expensive simulations or geometry reconstruction, *GRaF* lowers the barrier to widespread adoption while maintaining high performance. This is further supported by its improvement over NeRF2, achieving up to a 50.4% LPIPS reduction across simulated and real datasets.

E.2. Generalization to Unseen Layouts

Extending the generalization analysis in §5.4, we further evaluate the generalization ability of *GRaF* and *NeRF*^{2} (both trained on the Conference and Office layouts) on three additional and structurally distinct scenes:

- **Bedroom** (Figure 4a): 23.6 ft $\times$ 48.9 ft $\times$ 20.2 ft, with spatial spectra collected at 2700 transmitter locations.
- **Outdoor** (Figure 4b): 158.5 ft $\times$ 154.6 ft $\times$ 33.5 ft, with spatial spectra collected at 3082 transmitter locations.
- **RFID (Real-World)**: described in §5.1, covering 76.5 ft $\times$ 81.9 ft $\times$ 9.6 ft.

These three settings introduce diverse spatial characteristics: the *Bedroom* scene captures confined, multipath-rich indoor

Table 1. Performance on bedroom scene.

Model	MSE↓	LPIPS↓	PSNR↑	SSIM↑
KNN-DL	0.389	0.550	4.08	0.140
NeRF^2	0.429	0.575	3.68	0.160
GRaF	0.114	0.390	9.43	0.204

Table 2. Performance on outdoor scene.

Model	MSE↓	LPIPS↓	PSNR↑	SSIM↑
KNN-DL	0.525	0.750	2.82	0.120
NeRF^2	0.851	0.942	0.706	0.119
GRaF	0.208	0.591	6.83	0.148

Table 3. RFID performance without fine-tuning.

Model	MSE↓	LPIPS↓	PSNR↑	SSIM↑
KNN-DL	0.716	0.811	1.49	0.107
NeRF^2	0.927	0.958	0.319	0.072
GRaF	0.351	0.688	4.54	0.107

Table 4. RFID performance after fine-tuning.

Model	MSE↓	LPIPS↓	PSNR↑	SSIM↑
KNN-DL	0.085	0.326	10.73	0.518
NeRF^2	0.107	0.347	10.12	0.492
GRaF	0.049	0.239	15.97	0.618

propagation; the *Outdoor* scene represents a large-scale, open environment with sparse reflectors; and the *RFID* dataset introduces real-world noise, hardware variability, and temporal effects. This evaluation demonstrates the robustness of both models when applied to previously unseen environments beyond their training domain. KNN primarily serves to validate our interpolation theory; thus, we focus on reporting results for KNN-DL, *NeRF*^{2}, and *GRaF*.

Analysis. Both *GRaF* and *NeRF*^{2} exhibit degraded synthesis quality when evaluated on scenes that differ significantly from the training layouts, such as environments with unfamiliar furniture or large open spaces. The degradation is more severe for *NeRF*^{2}, which collapses into random predictions. In contrast, *GRaF* remains functional, generating spectra that preserve meaningful propagation cues, although its performance also declines. As demonstrated in A.2, the effectiveness of neighboring spectrum weighting is sensitive to transmitter positions. In unfamiliar layouts, these weights become more dynamic and difficult to estimate without explicit scene geometry, leading to reduced synthesis fidelity. KNN-DL also suffers a performance drop under layout shifts, similar to *NeRF*^{2}, but still outperforms it by leveraging neighboring spectra. Yet, KNN-DL underperforms compared to *GRaF* due to its reliance on scene-specific learned weights, which generalize poorly.

Despite the drop in synthesis fidelity on unseen layouts, the spatial spectra generated by *GRaF* remain valuable for downstream applications. A key use case is wireless localization, where spatial spectra serve as features to train localization models. For instance, in wireless localization (Section Case Study: Angle of Arrival (AoA) Estimation), they help initialize learning-based localization models more effectively than random initialization.

Fine-Tuning. *GRaF* 's generalization capability improves further with limited fine-tuning. We fine-tune KNN-DL, *GRaF*, and *NeRF*^{2}, each pretrained on the conference room layouts, using 20% of the training data from the RFID dataset. While all models benefit from this fine-tuning, *GRaF* achieves the best performance. This advantage stems from its ability to retain propagation priors learned during pretraining, requiring only the refinement of how it integrates neighboring spectra in the new scene. In contrast, *NeRF*^{2} needs to relearn voxel-level attributes from scratch, which limits its adaptability. Similarly, KNN-DL struggles to generalize via fine-tuning under limited data, as it lacks the embedded propagation modeling that enables *GRaF* to transfer effectively across scenes.

E.3. Parameter Study: Number of Neighbors

This section deepens the analysis behind Table 2 (in §5.3 of the main manuscript) by examining how the number of neighbors L influence *GRaF* 's zero-shot spectrum synthesis quality. All evaluations follow the same cross-scene protocol as in §5.3: one model is trained on *ConferenceV1* and tested on *ConferenceV2/V3*, and another is trained on *OfficeV1* and tested on *OfficeV2/V3*, with metrics averaged across all four unseen-scene configurations. All experiments use the 2.412 GHz band (wavelength $\lambda \approx 0.124$ m) and the same settings as in the main manuscript.

For each test transmitter location P, we construct a ranked pool of candidate neighbors by computing Euclidean distances to all training transmitters in the corresponding scene and sorting the points by increasing distance. During model training (as described and conducted in §5.3 of the main manuscript), we randomly sample $L \in [3, 10]$ neighbors for each transmitter to promote robustness to varying neighborhood densities; the Transformer thus learns to fuse an adaptively sized set of neighbors through cross-attention. At test time, we freeze the trained model and sweep over different

Table 5. Sensitivity to the number of neighbors L.

L	MSE↓	LPIPS↓	PSNR↑	SSIM↑
2	0.0538	0.274	17.98	0.658
4	0.0417	0.224	20.77	0.699
6	0.0399	0.221	20.88	0.704
8	0.0403	0.219	20.86	0.701
10	0.0396	0.220	20.90	0.709

values of L in order to analyze how the averaged performance in Table 2 emerges under different neighbor counts.

Number of neighbors L. Table 5 shows the effect of L on spectrum synthesis quality. We fix the radius to $\delta{=}10\lambda$ and select the first L neighbors from the sorted candidate list for each test location P. Because the model was trained with $L \in [3, 10]$, performance drops noticeably when L falls below this range. With $L{=}2$, PSNR decreases to 17.98 dB and both MSE and LPIPS degrade substantially. Within the trained interval ($L{=}4$–10), performance remains stable and exhibits fluctuations of approximately ± 0.1–0.2 dB in PSNR. These small variations reflect scene-dependent geometry. Cluttered regions may benefit modestly from additional neighbors because of richer multipath diversity, while open regions saturate earlier. Overall, the plateau across $L{=}4$–10 demonstrates the model's robustness to neighbor count and explains how the averaged performance reported in Table 2 arises across diverse test conditions.

E.4. Computation Overhead

Table 6 summarizes the model size and per-spectrum inference time of *NeRF*^{2} and *GRaF*. As expected, *GRaF* has a larger model footprint (478.6 MB vs. 8.3 MB) and a longer per-spectrum inference time (1.95 s vs. 0.43 s), primarily due to its geometry-aware Transformer and neural ray tracing modules. Unlike *NeRF*^{2}, however, *GRaF* does not require retraining for every new scene. At inference time, neighboring spectra are retrieved directly from the training split of the target scene, incurring no additional data-collection overhead.

Table 6. Comparison of model size and inference time.

	NeRF^2	*GRaF*
Model size (MB)	8.3	478.6
Inference time (s)	0.43	1.95

Although *GRaF* incurs higher per-sample inference overhead, both *NeRF*^{2} and *GRaF* scale linearly with respect to the number of synthesized spectra, making them suitable for large-scale generation tasks. Critically, when considering the *end-to-end* computational cost, including training and inference across multiple scenes, *GRaF* demonstrates higher efficiency. To illustrate this, we compute the total cost required to synthesize 32,625 spectra across the six scenes used in our experiments:

• ***NeRF*^{2}:** Each of the six scenes requires scene-specific training (10 hours per scene), totaling 60 hours. Inference adds an additional 3.9 hours (0.43 s per spectrum), leading to a total cost of *63.9 hours*.

• ***GRaF*:** A single model is trained once on layout-similar scenes in 15 hours. The inference phase requires 16.2 hours (1.79 s per spectrum), for a total of *31.2 hours*.

Overall, *GRaF* reduces end-to-end computation time by *approximately 51%* compared with *NeRF*^{2}. As the number of layout-similar scenes grows, these gains become even more pronounced because *GRaF* eliminates the need for repetitive scene-specific retraining. Finally, we acknowledge that *GRaF* 's per-spectrum inference latency is higher than that of *NeRF*^{2}. To mitigate this, future work may incorporate *3D Gaussian Splatting*, a technique widely used in optical NeRFs to accelerate rendering by replacing dense voxel sampling with adaptive, compact Gaussian primitives.

E.5. Impact of RF Frequency Bands

GRaF performs well when trained and tested within the same RF frequency band, but it does not generalize across different bands. This behavior is expected because wireless propagation is fundamentally *frequency-dependent*: reflection strength, diffraction angles, penetration depth, and scattering characteristics all vary with wavelength. As a result, the spatial spectrum at one frequency band cannot be reliably inferred from measurements at another. In practice, wireless communication systems are intentionally engineered to operate within *specific, standardized frequency bands*. Each band corresponds to distinct antenna designs, hardware front-ends, propagation characteristics, and communication protocols. Examples include:

• **1.8 GHz (4G LTE):** Optimized for wide-area coverage with relatively strong penetration. Devices use antenna and RF chains tailored to mobile broadband protocols such as OFDMA and SC-FDMA, enabling stable performance across large outdoor regions.

• **2.4 GHz (WiFi / WLAN):** Designed primarily for indoor deployments. Propagation is more sensitive to multipath and clutter from walls and furniture, and IEEE 802.11 protocols focus on robust medium access and interference mitigation in dense indoor environments.

• **3.0 GHz (5G NR):** Used in mid-band 5G with advanced hardware such as beamforming arrays and massive MIMO. Propagation exhibits reduced penetration and stronger directionality, and the 5G NR standard leverages these properties for high-capacity, low-latency communication.

These bands differ not only in propagation physics but also in system-level design: antenna geometries, transceiver circuitry, channel coding, and deployment conditions all diverge across frequencies. Consequently, a model trained at one frequency band cannot meaningfully generalize to another because the underlying spatial spectrum statistics shift with wavelength. For this reason, *GRaF* is intentionally designed for *per-band* spatial spectrum synthesis, which mirrors how real-world

wireless systems operate. In practice, building separate models for each frequency band is both realistic and necessary for producing physically accurate RF spectrum predictions.

F. Discussion

F.1. Rationale for Using Neighboring Spectra

While *GRaF* conditions on L sparse nearest–neighbor spectra at inference, this does not contradict our generalization claim. *GRaF* generalizes in the standard sense established by generalizable NeRFs: the model is trained *once* on a set of scenes and then directly applied to entirely new scenes *without any per-scene retraining or optimization*. The use of sparse neighbors is not a limitation, but a principled design choice mandated by our RF Spatial Spectrum Interpolation Theorem (Theorem 1), which proves that the spectrum at a target location can be approximated using spectra from geographically proximate transmitters with a *quadratically bounded* error. This behavior is inherent to RF propagation physics, where spatial smoothness and local continuity are well established (*e.g.*, path loss, diffraction, and multipath exhibit coherent local variations).

From a practical standpoint, obtaining L sparse neighbors is lightweight and realistic in real RF deployments. Modern WiFi/6G environments already contain multiple active transmitters, and sparse measurements can be obtained via quick walk-by scans, automated robot traversals, or even existing network infrastructure with negligible overhead. In contrast, *NeRF*^{2}/NeWRF require *dense* per-scene measurements (thousands of samples) and full per-scene model training, making them significantly more expensive to deploy. *GRaF* requires only a handful of conditioning samples and *no per-scene optimization*, reducing both measurement and computational cost by orders of magnitude.

Empirically, our cross-scene, cross-layout, and cross-environment experiments (Tables 2–3 and Supplementary Materials §E) demonstrate that *GRaF* consistently outperforms KNN, KNN-DL, and *NeRF*^{2} on unseen scenes despite using only sparse neighbors. These results confirm that conditioning on sparse local spectra *enables* practical and scalable generalization rather than limiting it. *GRaF* achieves strong generalization because it leverages (i) a scene-independent latent RF radiance field learned from diverse training scenes and (ii) physically grounded sparse conditioning aligned with RF interpolation theory, rather than relying on scene-specific overfitting or dense retraining.

F.2. Limitations and Possible Solutions

Despite the promising performance of our method, several limitations remain.

First, although *GRaF* generalizes well to scenes with moderately similar structures, its performance degrades when the new scene differs substantially from the training layouts. In such cases, a small amount of additional spectrum data may be required for scene-specific fine-tuning. This motivates future work on more expressive latent RF radiance fields that can better disentangle universal propagation behaviors from scene-dependent geometry.

Second, our current framework focuses on spatial generalization and does not explicitly address temporal dynamics. Real-world RF environments exhibit temporal variations caused by moving objects, human activity, or environmental changes. A promising direction is to extend *GRaF* with a deformable or time-aware latent RF radiance field that can capture temporal evolution, enabling continuous adaptation without full retraining.

Finally, *GRaF* assumes access to sparse neighboring spectra at inference, which is lightweight but still requires minimal on-site measurements. Future research may explore hybrid approaches that combine *GRaF* with predictive physics-based models or self-supervised priors, further reducing measurement requirements and enabling fully zero-shot deployment in unseen environments.

References

[1] Shenchang Eric Chen and Lance Williams. View Interpolation for Image Synthesis. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1993. 6

[2] Weng Cho Chew. *Waves and Fields in Inhomogenous Media*. John Wiley & Sons, 1999. 2

[3] Mozhdeh Gheini, Xiang Ren, and Jonathan May. Cross-Attention Is All You Need: Adapting Pretrained Transformers for Machine Translation. *arXiv preprint arXiv:2104.08771*, 2021. 10

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 10

[5] Joseph B Keller. Geometrical Theory of Diffraction. *Journal of the Optical Society of America*, 52(2):116–130, 1962. 2

[6] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 10

[7] R.G. Kouyoumjian and P.H. Pathak. A Unifm Geometrical Theory of Diffraction for an Edge in a Perfectly Conducting Surface. *Proceedings of the IEEE*, 62(11):1448–1461, 1974. 10

[8] MATLAB. Indoor MIMO-OFDM Communication Link Using Ray Tracing, 2023. [Online]. 5

[9] Alireza H Mohammadian, Vijaya Shankar, and William F Hall. Computation of Electromagnetic Scattering and Radiation Using a Time-Domain Finite-Volume Discretization Procedure. *Computer Physics Communications*, 68(1-3):175–196, 1991. 10

[10] Theodore S Rappaport. *Wireless Communications: Principles and Practice*. Cambridge University Press, 1996. 2

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Conference on Neural Information Processing Systems*, 30:6000–6010, 2017. 10