

# Generative Adversarial Networks for Accelerated Ray-Tracing in Indoor Wireless Channels

Michael P. Acquaviva

michael.acquaviva@mail.utoronto.ca

**Abstract**—We present a hybrid approach to modeling radio wave propagation in complex indoor environments. Our model combines ray-tracing, a deterministic method, with a generative adversarial network (GAN) to enhance predictions of received signal strength (RSS) coverage. By limiting the number of ray interactions considered in the ray-tracing step, our model drastically reduces the computational costs associated with high-fidelity ray-tracing. Unlike previous hybrid approaches, our model decomposes the RSS quantity into two components, a free-space component and a faded component, with the GAN predicting strictly the latter. We use Friis' transmission equation to model the free-space component, while the faded component is quantified by the *faded path gain* (FPG) term of the log-distance path loss model. Although mathematically, we expect the FPG term to be log-normally distributed, we do not force it to follow any particular distribution. On average, our model provides a reduction in mean runtime of 87%. Our model generalizes in three main areas: transmitter positions, frequencies, and environment geometries. In each of these cases, we demonstrate relative errors between predicted and ground truth data of 4.09%, 3.12%, and 3.77%, respectively. Our model presents lower absolute errors in each of the three areas of generalizability when compared to strictly ray-tracing considering a limited number of ray interactions.

## I. INTRODUCTION

The accurate and efficient modeling of indoor radio wave propagation presents many challenges due to the multipath effects present in such environments [1]. The ability to rapidly produce high-fidelity renderings of the received signal strength (RSS) in an environment would serve as an invaluable tool for communication systems designers. Such a tool could be used to determine the optimal placement for wireless access points (WAPs) [2], allowing designers to reduce costs by minimizing the number of antennas needed to adequately service large areas. Alternatively, accurate RSS predictions are essential in ensuring that human exposure to electromagnetic fields is constrained to safe thresholds [3].

In practice, both physical and empirical propagation models have been developed, however, both routes have their limitations. Empirical models often fail to generalize accurately to frequency, geometry, and material-specific details, requiring labour-intensive and expensive measurement campaigns. Conversely, physical models lack the computational efficiency to make them feasible in electrically large environments [4]. Despite the potential for accuracy in well-calibrated physical models, measurement campaigns are typical in the development of site-specific empirical propagation models [2].

In this article, we present a hybrid approach, using neural networks to overcome the computational challenges associated with physical modeling. Our specific focus is on ray-tracing (RT) [5], a physical model which is widely used in computer graphics to model the behaviour of light. This algorithm works by tracing the path of rays and their interaction with the environment, in the form of reflections, transmissions and diffractions [6]. For an accurate simulation, RT must consider many rays and ray-interactions, leading to high computational costs. At the expense of accuracy, we can greatly reduce the RT runtime by limiting the number of ray-interactions considered [7]; we refer to RSS maps produced in this way as *low-fidelity*. Our framework makes use of a generative adversarial network [8] (GAN) to map low-fidelity results, produced by quick RT simulations, to more accurate, *high-fidelity* renderings [9].

Considering fewer ray-interactions makes low-fidelity RT highly prone to errors, especially in regions which have a large number of obstructions between the receiver and transmitter. Figures 1 b) and 1 d) depict the results of low- and high-fidelity RT respectively. These images provide a map of the RSS at different points throughout the room. As can be seen in the renderings, low-fidelity RT greatly underestimates the RSS coverage in certain areas, shown in purple on the low-fidelity map. While low-fidelity RT is computationally practical, its product lacks in accuracy [10].

The approach we take differentiates from previous AI-based propagation models. While previous projects have attempted to train AI models using several case-specific features (e.g., the number of obstructions, the floorplan of the environment, and the presence of line-of-sight (LoS) / non-line-of-sight (nLoS) points) [11], our model approaches the problem as *image enhancement*, i.e., trying to decrease the error in the low-fidelity rendering we currently have. This approach has been used by Seretis et. al. [9], where it provided them with the ability to generalize maps of RSS, for a fixed transmitter position and geometry, to different frequencies and receiver distances. Our work builds off this model. In their work, Seretis et. al. used two-dimensional maps of RSS values produced by low-fidelity RT as input features into their GAN, with the high-fidelity counterparts serving as labels. For the purposes of this paper, we refer to GAN models such as this one, which predict RSS maps directly, as RSS-GAN.

In this paper, we hypothesize that training based on coefficients from the *log-distance path loss model* [12]

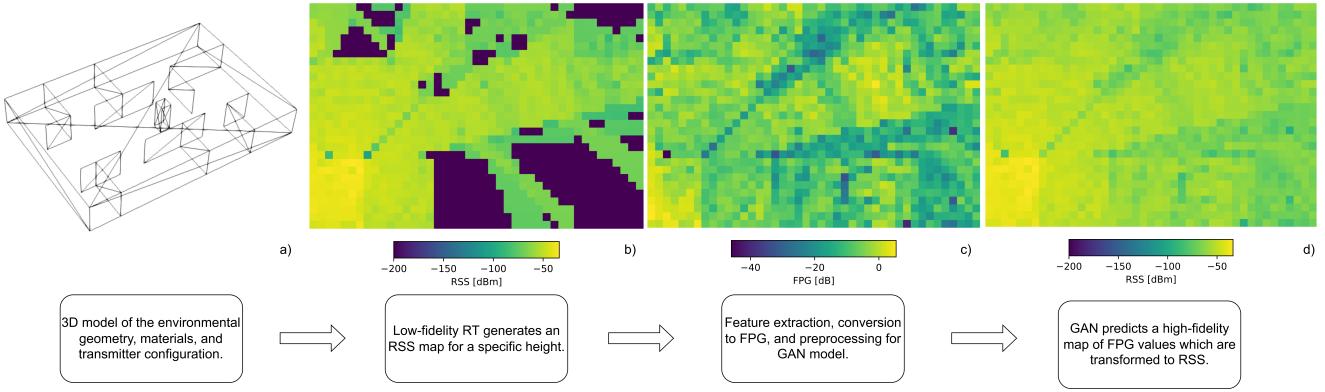


Fig. 1: A high-level depiction of our proposed model’s operation. a) Depicts the three-dimensional environment in stereolithography format. b) Shows an RSS map produced using low-fidelity RT. c) Shows a prediction of the environment’s FPG map. d) Shows the predicted RSS map after being mapped from FPG values. For the high-fidelity RSS map, we considered 6 reflections, 4 transmissions, and 1 diffraction. For the low-fidelity map, we considered 1 of each interaction.

would improve the accuracy of GAN predictions over RSS. The model is explained mathematically in Section II; it involves subtracting, from the observed powers, the powers which would result strictly from free-space transmission. This operation yields the *faded path gain* (FPG), which is easily mapped back to RSS. Our hypothesis is grounded in the following: FPG values remove the propagation effects caused by free-space transmission and therefore, they are a direct quantification of the multipath effects present in indoor channels. Our process effectively decomposes the RSS value into two components: the free-space path loss (FSPL), which is easily modeled as a function of distance; and the FPG, which is a function of the geometry. Since we wish to generalize over a wide variety of input geometries, the use of FPG values are purely and directly related to the input variable. We therefore expect a neural network to learn the mapping between the geometry and FPG more efficiently than mapping to RSS, which has additional dependencies. We call this novel approach FPG-GAN, as it indirectly predicts RSS maps by way of FPG maps.

The structure of the article is as follows. In Section II, we begin by providing an overview of the log-distance path loss model which we use to construct features for our GAN. In Section III, we introduce our GAN along with the data generation, preprocessing, training, and evaluation methods. In Section IV, we present the results of FPG-GAN’s predictions concerning three main areas of generalizability: transmitter position, frequency, and environment geometry. Section V serves as a discussion, wherein we compare the performance of training on FPG values, FPG-GAN, to the traditional method, RSS-GAN; we also explore some phenomena we observed arising from variable frequencies in multipath channels. Finally, Section VI concludes the article with a summary of our findings.

## II. THE LOG-DISTANCE PATH LOSS MODEL

This section provides an overview of the log-distance path loss model which we use to construct features for our GAN.

This model provides an stochastic solution to a faded indoor multipath propagation.

### A. Mathematical Formulation

Fading channels arise due to variable attenuation within a environment. Many factors influence the extent of this attenuation, including the frequency of transmission, the relative position of transmitting antennas to environmental obstacles, and even the time elapsed from the onset of transmission [13]. Due to the complexity of these phenomena, it is often efficient to model fading as a random process [14]. The log-distance path loss model attempts to model this process as follows:

$$P_r^{[dB]}(r) = P_r^{[dB]}(r_0) - 20 \log_{10} \left( \frac{r}{r_0} \right) + X^{[dB]} \quad (1)$$

where  $P_r^{[dB]}$  is the decibel RSS,  $r$  is the receiver-transmitter distance, and  $X^{[dB]}$  is a random variable expressing the FPG in decibels [12]. The term  $r_0$  is a reference distance in the far-field region, often selected to be 1 m.

We can interpret the FPG term,  $X^{[dB]}$ , as a source of random fluctuations around the free-space transmission solution. To this effect,  $P_r^{[dB]}(r_0)$  is given by Friis’ transmission equation:

$$P_r^{[dB]}(r_0) = G_t^{[dB]} + G_r^{[dB]} + P_t^{[dB]} + 20 \log_{10} \left( \frac{\lambda}{4\pi r_0} \right) \quad (2)$$

In Eq. (2),  $G_t^{[dB]}$  and  $G_r^{[dB]}$  are the transmitter and receiver directive gains, respectively,  $P_t^{[dB]}$  is the decibel transmitted power, and  $\lambda$  is the wavelength. The ideal WAP is an isotropic radiator. While such a source of radiation is theoretically unattainable, most WAP transmitters can be approximated as having zero directive gains. Using the standard  $r_0 = 1$  m, this allows us to combine (1) and (2) as:

$$X^{[dB]} = P_r^{[dB]} - P_t^{[dB]} + 20 \log_{10}(4\pi) + 20 \log_{10}(d_E) \quad (3)$$

where we define the *electric distance*,  $d_E \triangleq \frac{d}{\lambda}$ .

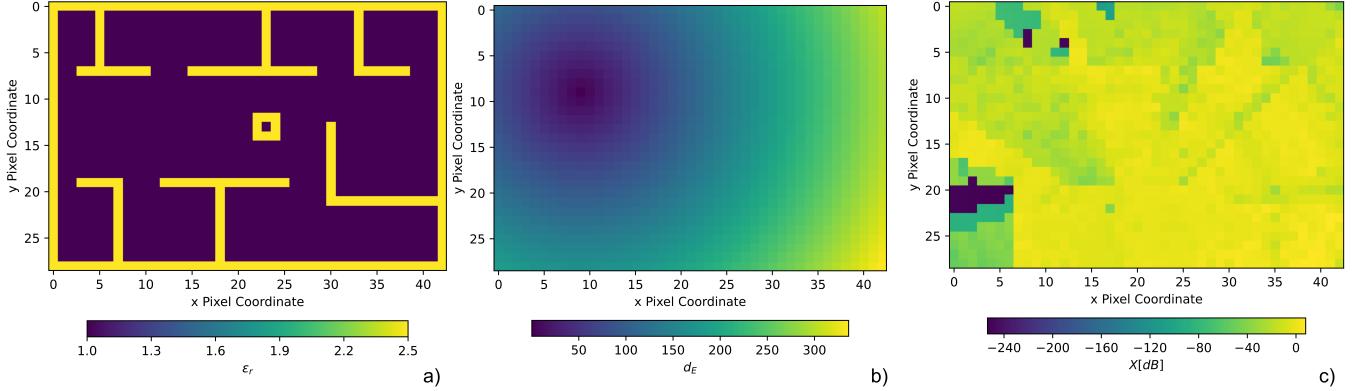


Fig. 2: The three input features for the FPG-GAN model. a) is a 2D representation of the environment floorplan. b) is a map of the transmitter-receiver distances in wavelengths. c) is a map of FPG values produced by low-fidelity RT.

This model gives us a formulation for the FPG value,  $X^{[dB]}$ , in the decibel scale which we use to quantify the level of fading attributed to the geometry. The last two terms in (3) comprise the FSPL; when we add this quantity and subtract  $P_t^{[dB]}$ , this is analogous to removing all effects caused by the free-space component of transmission from the RSS. Mathematically, we expect the FPG quantity to approximately follow a Gaussian distribution [15], however our model does not constrain  $X^{[dB]}$  to take on any particular distribution.

### B. Constructing Features

We argue that the combination of FPG values,  $X$ , and  $d_E$  provide a *natural* set of features for an AI model, allowing us to better generalize to new frequencies, geometries, and transmitter positions. Firstly, free-space transmission is trivial to model via (2), therefore we are strictly interested in learning the *gain* attributed to fading channels. The addition of the FSPL in (3) achieves this; creating a random variable,  $X$ , independent of free-space transmission effects. Secondly,  $d_E$  encodes two important propagation quantities: the frequency and receiver-transmitter distance. The function of  $\lambda$  is to scale the distance – making environments *appear* effectively larger for shorter wavelengths. In effect, this quantity should aid in frequency generalizability.

Our model uses three regressor variables to predict FPG values accurately. They are as follows:

- 1) a two-dimensional floorplan of the environment;
- 2) a two-dimensional map of  $d_E$ ;
- 3) and a two-dimensional map of FPG values produced from low-fidelity RT.

Illustrations of these regressors are shown in Figure 2; they are passed to the FPG-GAN model as individual 2D planes.

### III. METHODS

In this section, we describe our approach to building, testing, and evaluating GAN-based RSS-map enhancement models. As a matter of course, we discuss the following challenges:

- 1) constructing appropriate datasets that allow our model to generalize to different receiver positions, frequencies, and environmental geometries;
- 2) choosing a GAN architecture that is efficient to train and run, while also providing accurate predictions of the true RSS (by way of FPG);
- 3) choosing preprocessing steps which compliment the GAN model and the goal of generalizability;
- 4) establishing cost functions that allow our GAN to *learn* to model propagation optimally;
- 5) selecting a set of evaluation metrics which allow us to compare the performance of our model to others. Namely, this involves comparing the performance of FPG-GAN (FPG-based regression) to RSS-GAN (RSS-based regression).

#### A. RT Dataset Generation

To demonstrate each of the three areas of generalizability, we produce three distinct datasets using RT, each containing 300 pairs of low- and high-fidelity RSS maps. To produce these maps, we use environments of the same external dimensions, 22 m x 15 m x 3 m, saved in stereolithography (STL) format. For simplicity, we assume that each surface is made of the same material, drywall. We assumed a thickness of 0.02 m, a relative permittivity of 2.5, and a conductivity of 0.3 S/m. These values closely resemble materials used in practical office environments [2].

To produce RSS maps, we begin by tiling the geometry with a plane grid of receiving antennas, located at a height of 1.5 m. Each point is responsible for covering an area of 0.5 m x 0.5 m, resulting in a 43 x 29-pixel representation of the space in 2D, with the receiving antennas located in the center of each pixel. All transmitting antennas are placed at a height of 0.5 m. We then use an in-house RT program [16] to produce RSS maps at both of the desired levels of fidelity [9].

Table I shows the number of each type of ray-interactions we use to produce RSS maps at each level of fidelity, along with the computational runtime on six Intel Xenon X5680 cores.

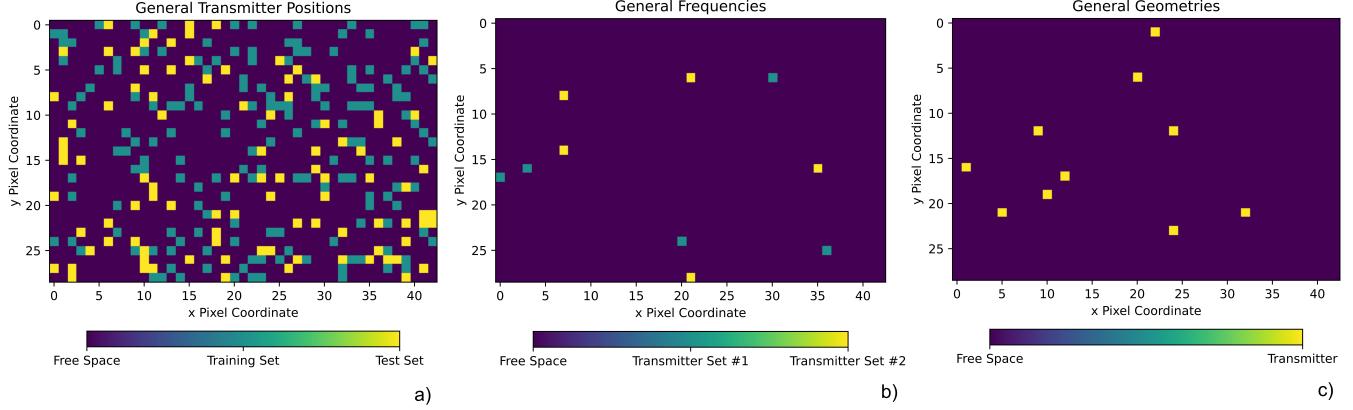


Fig. 3: The locations of transmitting antennas for each of our three datasets.

Below, we elaborate on the construction of each of our three datasets.

1) *General Transmitter Positions*: In this dataset, we produce RSS maps by moving the transmitting antenna throughout a fixed geometry. Of the 1247 pixels on our 2D map of the environment, we sample 300 transmitter positions from a uniform distribution. We run the RT simulations on each of these points using a fixed frequency of 5.3 GHz, which is a typical value for wireless LAN applications. We then divide the dataset in two, with  $\frac{2}{3}$  of the points used for training our model, and  $\frac{1}{3}$  used for testing. The geometry we use is illustrated in Figure 2 a).

2) *General Frequencies*: In this dataset we produce RSS maps by considering 60 equally-spaced microwave frequencies, from 3 to 30 GHz. We use the same geometry as in the general transmitter dataset. For the lower 30 frequencies, we consider a set of five randomly sampled transmitter positions and five different positions for the remaining 30. In maintaining the same test-train split as in the previous case, every third frequency in our dataset is moved to a testing set.

3) *General Geometries*: In this dataset, we produce RSS maps by considering 30 different geometries, i.e., different arrangements of interior walls within a fixed space. To create our dataset, we fix the frequency at 5.3 GHz and randomly select 10 fixed transmitter points, which we use to ray-trace each geometry.

The geometries we use are the same as those used in [2]. 14 of these are classified as *simple*, and 16 as complex. Images of the geometries we used can be found in Appendix A. To construct our training and test sets, we divide the datasets by geometry. Our test set contains 6 geometries from the *complex* class, as those most closely resemble the *office-like* environments we are aiming to model, shown in Figure 4.

Fidelity	Reflections	Transmissions	Diffractions	Time [s]
Low	1	1	1	41
High	6	4	1	305

TABLE I: The number of each ray-interaction we considered, along with the mean runtime, for both RT fidelity levels.

### B. GAN Architecture

As their name suggests, GANs consist of two *competing* neural networks which ultimately converge to find the lowest-cost solution [8]. These models are referred to as a *generator* and a *discriminator*.

A flowchart of our FPG-GAN model is shown in Figure 5. We begin by inputting our geometry, transmitter position, and other parameters,  $x_{RT}$ , into the RT solver. This operation gives us a set of low- and high-fidelity RSS maps,  $\mathbf{P}_{r_{LF}}$  and  $\mathbf{P}_{r_{HF}}$ , respectively. By way of 3, we map these to their corresponding FPG values. The low-fidelity FPG map,  $\chi_{LF}$ , is then passed to the U-Net generator which attempts to predict a more accurate depiction of the FPG map,  $\hat{\chi}$ . This prediction is guided by additional input features, namely the environment floorplan,  $\mathbf{E}$ , and the electric distance map,  $\mathbf{d}_E$ . Again, using (3), we map the predicted FPG values to a prediction of the RSS map,  $\hat{\mathbf{P}}_r$ . In the process of training our model, we alternate in passing the generator output,  $\hat{\chi}$ , and the high-fidelity FPG map,  $\chi_{HF}$ , to the discriminator model. This model tries to make a prediction,  $z \in (0, 1)$ , of whether the sample in question came from the generator or high-fidelity RT. We use the  $z = 0$  label to denote a generated result and  $z = 1$  to denote a result from high-fidelity RT. We train the generator using back-propagated feedback from the discriminator,  $L_1(z)$ , to force  $\hat{\chi}$  to *look* more like  $\chi_{HF}$ . We also employ a second loss function in training the generator,  $L_2(\hat{\mathbf{P}}_r, \mathbf{P}_{r_{HF}})$ , to account for quantitative errors.

The remainder of this section introduces the specific architectures we employ for both the generative and discriminative models. Both are both constructed in the PyTorch framework.

1) *U-Net Generator*: For the generative model, we employ a deep residual U-Net architecture [17]. Figure 6 provides a detailed depiction of the different model layers.

The input layer consists of three planes containing the features listed in Section II-B. The input size of these planes is 32 x 48 px. To obtain this size, we zero-pad the bottom and right edges of the 29 x 43 px. features. In every tier of the U-Net, we make use of convolutions with a kernel size of 3, a stride of 1, and same padding. Next, we make use of convolutional residual blocks – structures where the input is

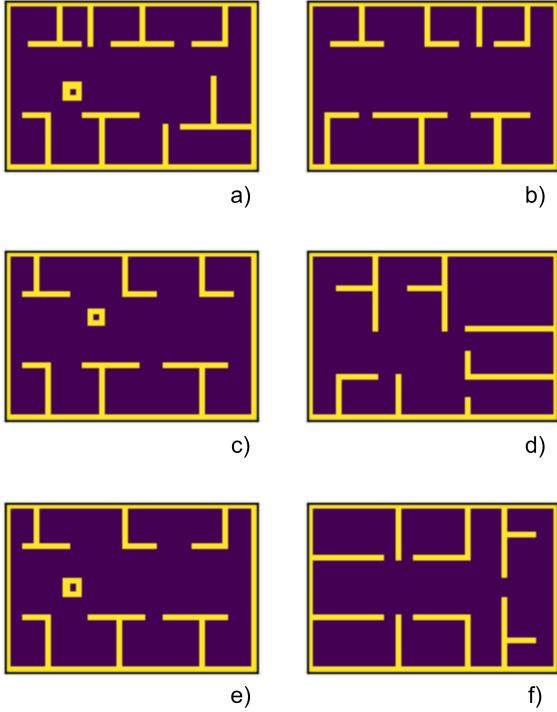


Fig. 4: The floorplans belonging to the testing set.

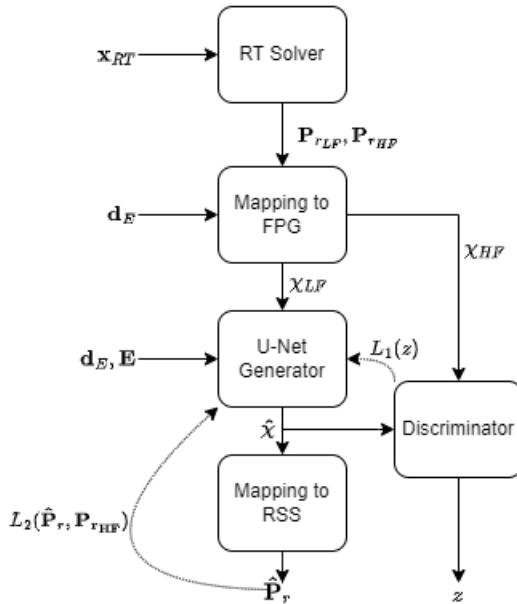


Fig. 5: A detailed flowchart of the FPG-GAN model.

added back to the output. Residual blocks have been shown to accelerate learning for AI models since only the *difference* between a layer's input and output must be learned, and not the full mapping [18]. For tiers on the left side of the network, down-sampling is performed through a max pool operation, with a kernel size of 2. In contrast, for tiers on the right half, we perform up-sampling using a transpose convolution with a kernel size of 3 and a stride of 2. Each layer in the network

uses batch normalization (BN) [19] and a rectified linear unit activation function (ReLU). The output layer is a single 32 x 48 px. plane whose goal is to predict FPG maps. We ignore the extra pixels on the bottom and right to obtain the desired 29 x 43 px. size.

2) *Discriminator*: For the discriminative model, we employ a convolutional binary classification model. Figure 7 provides an illustration of the layers involved in the architecture.

The input layer of the discriminator accepts a 32 x 48 px. plane representing an FPG map. There are 4 convolutional layers in the discriminative model, each having a kernel size of 3, a stride of 2, and zero-padding of 1. These layers use the leaky-ReLU (LReLU) activation function. In the latter 3 layers, batch normalization (BN) is employed and the number of filters are doubled through each layer. After the last convolutional layer, all filters are flattened into a one-dimensional array of size 1536 and passed into a fully connected (FC) layer. Finally, a sigmoid activation function is employed, constraining the output to  $z \in (0, 1)$  [20].

### C. Preprocessing

In this section, we discuss the operations taken between the output of data from RT and the input into the GAN model, either for training or evaluation. Preprocessing serves three main purposes in our work:

- 1) to ensure our data is in a form which could be easily discerned by the GAN model;
- 2) to optimize our model's ability to *learn* the complex properties of wireless propagation;
- 3) and, in some cases, to generate more data from the data we already have.

Below we introduce our preprocessing steps in sequential order.

1) *Data Normalization*: Data normalization is a common practice for constructing AI models, enhancing their performance and training stability. The goal of normalization is to ensure that all of the features are on a similar scale, i.e., there is no significant numerical bias towards one feature over another. After mapping RSS values to FPG, we find the mean,  $\mu$ , and variance,  $\sigma^2$ , of all pixel values across the entire training subset of our dataset. Using these statistics, we then transform each pixel value according to the following:

$$\tilde{\chi}^{[dB]} = \frac{\chi^{[dB]} - \mu}{\sigma} \quad (4)$$

This transformation yields  $\tilde{\chi}^{[dB]}$  values which have zero mean and unit variance. We employ the same normalization mapping for the quantity  $d_E$ . For the floorplan input, we achieve similar scales by allowing wall pixels to take on a value of 1 and free space pixels to take on a value of -1.

2) *Linear-Scale Mapping*: Since the generator output layer makes use of a ReLU activation function, it can only yield positive values. To account for the possibility of negative values in the decibel scale, we apply the following mapping:

$$\tilde{\chi} = 10^{\frac{\tilde{\chi}^{[dB]}}{10}} \quad (5)$$

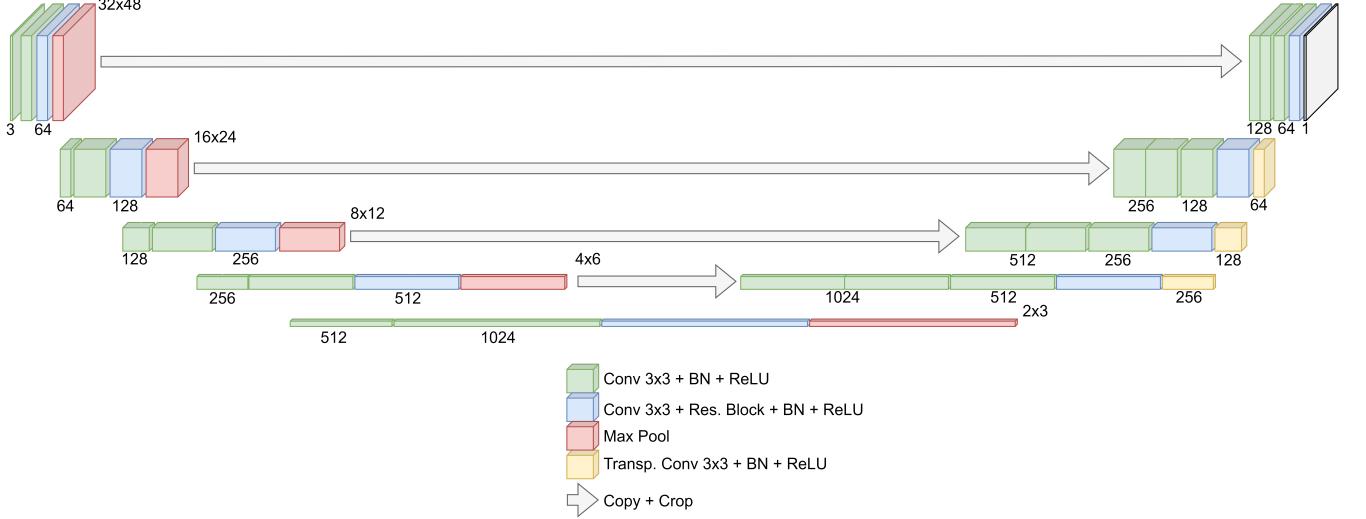


Fig. 6: Deep residual U-Net generator architecture.

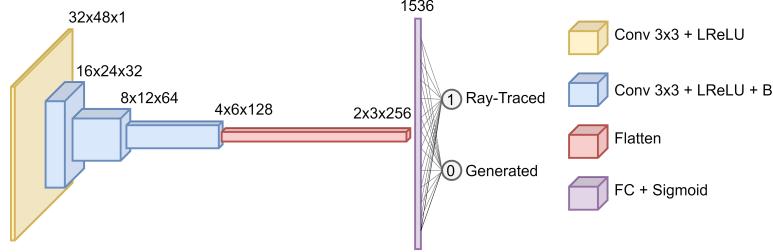


Fig. 7: Convolutional binary classification discriminator.

The same mapping is applied to normalized  $d_E$  values, but not the floorplan input. This mapping also has implications on the distribution of  $\tilde{\chi}$ . In theory, a lognormal distribution is expected [15], however in our model, we only place the constraint that all the probability mass must lie in  $\tilde{\chi} \geq 0$ . After being outputted by the model, we convert values back to the decibel scale.

*3) Dynamic Noise Addition:* A common challenge faced when training GANs is experiencing *mode collapse*, i.e., having the generator produce outputs with little variation across multiple inputs. This problem stems from the discriminator predicting the correct label throughout the majority of training [21]. To address this challenge, we add random noise to the discriminator inputs, sampled from a lognormal distribution. We use a distribution such that the corresponding Gaussian mean and variance are 0 and 1, respectively. The amplitude of noise added is regulated by a dynamic noise coefficient,  $\zeta$ , which we initially set to a value of 5. The added noise, therefore, is the product of  $\zeta$  and the values sampled from the noise distribution.

*4) Data Augmentation:* To encourage the learning of propagation through different geometries, we employ augmentation techniques to increase the size of the generalized geometry dataset. To accomplish this, we take mirror images in the horizontal and vertical axes, as well as a 180° rotation. In effect, this expands the number of geometries from 30 to

120.

#### D. Model Training

In training our model, we use a hybrid cost function. As is typical with GANs, we employ the Minimax loss function:

$$L_1 = \mathbb{E}_{\chi_{HF}} [\log(D(\chi_{HF}))] + \mathbb{E}_{\hat{\chi}} [1 - \log(D(\hat{\chi}))] \quad (6)$$

where  $D$  denotes the discriminator output for the given input. This loss function encourages the generator to produce maps which *fool* the discriminator. The second part of our hybrid loss function accounts for the error in the generator's predictions:

$$L_2 = \mathbb{E}_{\hat{\chi}} |\hat{\chi} - \chi_{HF}| \quad (7)$$

The overall cost function is a regularized sum of both  $L_1$  and  $L_2$ :

$$L = L_1 + \lambda L_2 \quad (8)$$

where  $\lambda = 100$  is a regularization constant. In training, we make use of the Adam optimizer with an initial learning rate of 0.01. We run the training loop, with a batch size of 25, for 125 epochs halving the learning rate and noise coefficient,  $\zeta$ , every 25 epochs. For the last 25 epochs, we set  $\zeta = 0$ .

### E. Evaluation

To evaluate the performance of our model, we make use of three main metrics. First, we consider the mean absolute error (MAE) between the GAN-predicted RSS maps and their high-fidelity counterparts in the testing set. This quantity, measured in dB, is given by Equation (7) and informs us on the pixel-wise accuracy of the GAN model.

Thus far, we have described predicting RSS maps by way of FPG values. To test our central hypothesis – i.e., whether mapping to FPG enhances the GAN’s ability to generalize in each of our three areas of interest – we reran each of our experiments, this time skipping the FPG mapping step. This provides us with a second model which predicts RSS values directly, analogous to previous works. We refer to the FPG-based model as *FPG-GAN* and the RSS-based model as *RSS-GAN* [9]. By looking at the percentage of instances in which FPG-GAN provides a lower MAE than RSS-GAN, we deduce a metric which indicates how superior or inferior our model is to the RSS-based approach.

Lastly, to evaluate FPG-GAN’s ability to generalize to each area of interest, we consider the percentage error between the probability distributions of the generated and high-fidelity FPG values. This metric is given by:

$$\epsilon = \frac{\int_{-\infty}^{\infty} \sqrt{(\hat{f}(\chi) - f_{HF}(\chi))^2} d\chi}{\int_{-\infty}^{\infty} \sqrt{(f_{HF}(\chi))^2} d\chi} \quad (9)$$

where  $\hat{f}(\chi)$  is the distribution of predicted FPG values and  $f_{HF}(\chi)$  is the distribution of high-fidelity FPG values. We compare this to the percentage error between the low- and high-fidelity FPG distributions to deduce the level of improvement FPG-GAN provides at fitting the high-fidelity RT data. Note, the denominator integrates to 1 as per the nature of PDFs and the numerator is akin to taking the absolute value of a difference of functions.

### IV. FPG-GAN RESULTS

In this section, we present the results of our FPG-GAN simulations for each of the three areas of generalizability. Specifically, we evaluate the model in terms of its MAE and percent difference between actual and generated distributions.

Figure 8 illustrates the percent reduction in MAE for test-set images before and after passing them through the trained FPG-GAN models. In 100% of the test cases, across each dataset, FPG-GAN provides a reduction in MAE – i.e., no maps are worse-off after being passed through the model. From the results in each of our experiments, we observe a positive correlation between the initial, low-fidelity MAE and the percent reduction in MAE. This means that our model compensates for the lack of accuracy moreso in samples which begin with large errors. The implications of this property are such that FPG-GAN maps a wide range of input (low-fidelity) MAEs to a relatively narrow range of output MAEs. This is a desirable property for regression models, as it allows for tight confidence intervals to be placed on the error of the output.

Table II presents the aggregate test-set results of each experiment; to deduce the values, we took the mean of the sample-wise metrics. In terms of the lowest MAE, FPG-GAN performed best in the geometry generalization case. In terms of mean absolute percentage error (MAPE) – a normalized version of MAE – the model performed best in the frequency generalization case. Hence, we can deduce from this result that our use of  $d_E$ , to express distances as a number of wavelengths, provides an effective method of scaling environments to generalize to different frequencies. Our results are notable, as they imply that the model can generalize to different geometries and frequencies to a better degree than it can to different transmitter positions – a task which we assumed originally to be more trivial.

We observe a clear reduction in MAE between the low-fidelity and predicted solutions in each of our experiments. Figure 9 illustrates how the percentage reduction in MAE is distributed across all of our test samples. In the general transmitter position and general frequency cases, the percentage reductions in MAE appear spread over a large range. We note that the power of our model to reduce the MAE of any particular sample is highly dependant on the initial MAE of the low-fidelity input – for samples which are already low in MAE, our model provides modest improvements. Nonetheless, our model enhances the accuracy 100% of the time. In the general geometry case, we see MAE reductions skewed heavily towards the higher end. In this case, we also observe the lowest MAE. From these findings, we can deduce that our model is most efficient at learning when trained on multiple geometries. In each of the areas of generalizability: transmitter positions, frequencies, and geometries, FPG-GAN demonstrates average reductions in MAE of 61%, 47%, and 84%, respectively.

Figure 10 illustrates the distributions of the low- and high-fidelity FPG values, as well as the FPG-GAN fit. In terms of the goodness of fit of the generated distributions, the general frequency case provides, by far, the best fit to the high-fidelity distribution. This is particularly notable as the frequency case, shown in Figure 10 b), depicts this curve as the least Gaussian in nature, compared to the other cases. Unlike what theory suggests [12] [15], we observe curves which are not exactly Gaussian in nature. In fact, all of the curves demonstrate a degree of bimodality, which is most pronounced in the general frequency case. The phenomenon causing this bimodality will be explored in Section V. Nonetheless, the FPG-GAN generated curves fit the high-fidelity curves better than the

Experiment Case	LF-MAE [dB]	$\hat{X}$ -MAE [dB]	$\hat{X}$ -MAPE [%]	$\epsilon_{LF}$ [%]	$\epsilon_{\hat{X}}$ [%]
Position	12.14	2.49	4.09	38.42	16.90
Frequency	7.68	2.34	3.12	17.05	4.91
Geometry	26.15	2.28	3.77	38.42	16.78

TABLE II: A summary of the aggregate results of the FPG-GAN simulations in each experiment. We present both the low-fidelity (LF) and predicted ( $\hat{X}$ ) MAE and  $\epsilon$ , as well as the mean absolute percentage error (MAPE) of the output.

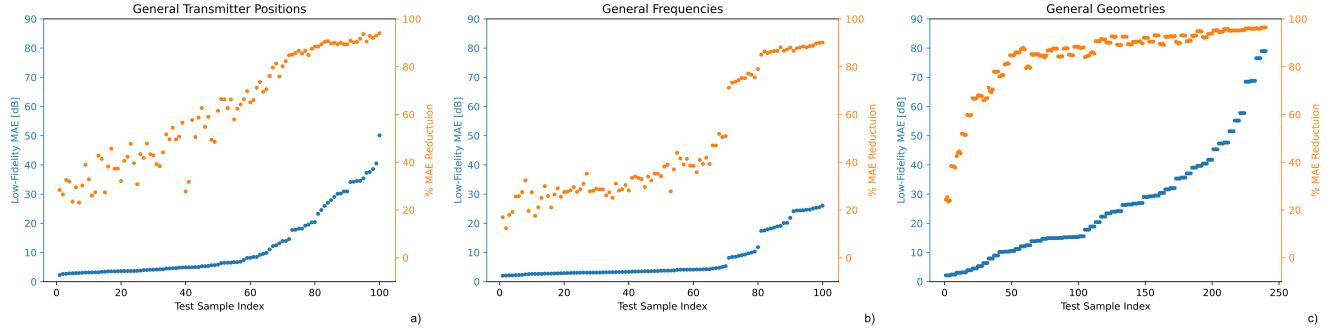


Fig. 8: The image-wise percent reduction in MAE for FPG-GAN in each area of generalizability. The left axis and blue points depict the starting, low-fidelity MAE. The right axis and orange points depict the percentage reduction in MAE.

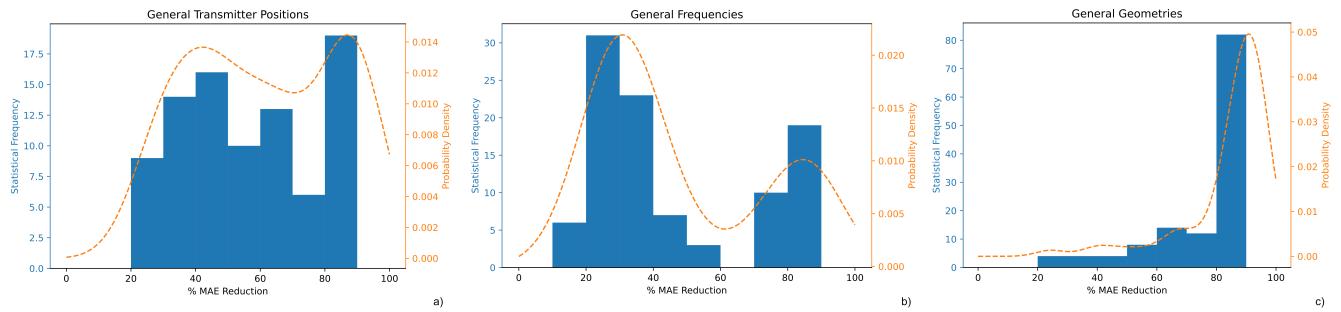


Fig. 9: Histogram plots of the percent reduction in MAE for each area of generalizability.

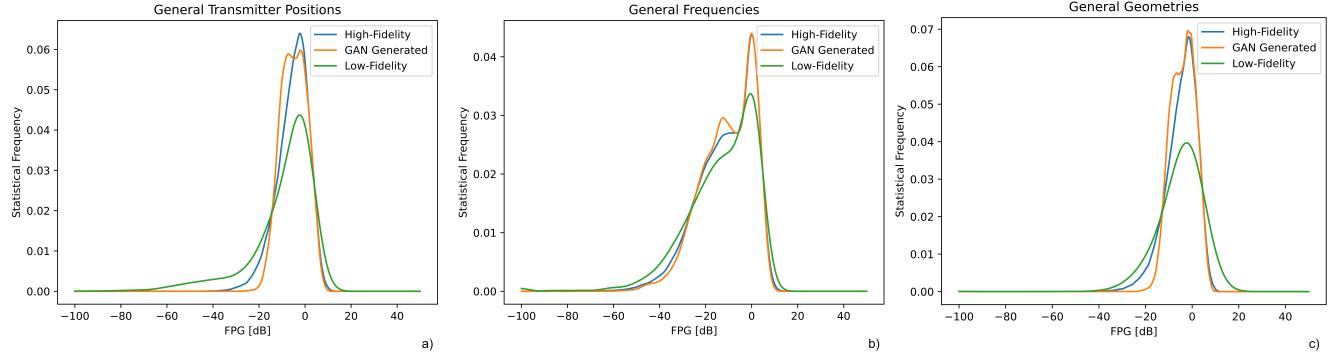


Fig. 10: The distributions of  $X$  for low- and high-fidelity maps in each area of generalizability, as well as FPG-GAN’s fit of the high-fidelity distribution.

low-fidelity curves in all three cases; this can be deduced both visually and via the  $\epsilon$  metric.

## V. EVALUATION AND DISCUSSION

In this section, we evaluate the FPG-GAN model against RSS-GAN to test our hypotheses. We also provide a deeper analysis of the existence of bimodality in the general frequency FPG distribution. We use the findings from this analysis to further discuss possibilities for the future improvement of the FPG-GAN model.

### A. Performance Comparison to RSS-GAN

After rerunning our experiments using the RSS-GAN model, we noted that in all three cases the FPG-GAN model yielded a lower aggregate MAE. This implies that the overall error associated with FPG-GAN is lesser than that associated with RSS-GAN. In the general geometry case, both models practically delivered the same level of accuracy (deviating by only 0.2%). With regards to instances in which FPG-GAN outperformed RSS-GAN in the testing sets, it did so by large margins in the general position and frequency cases, but was

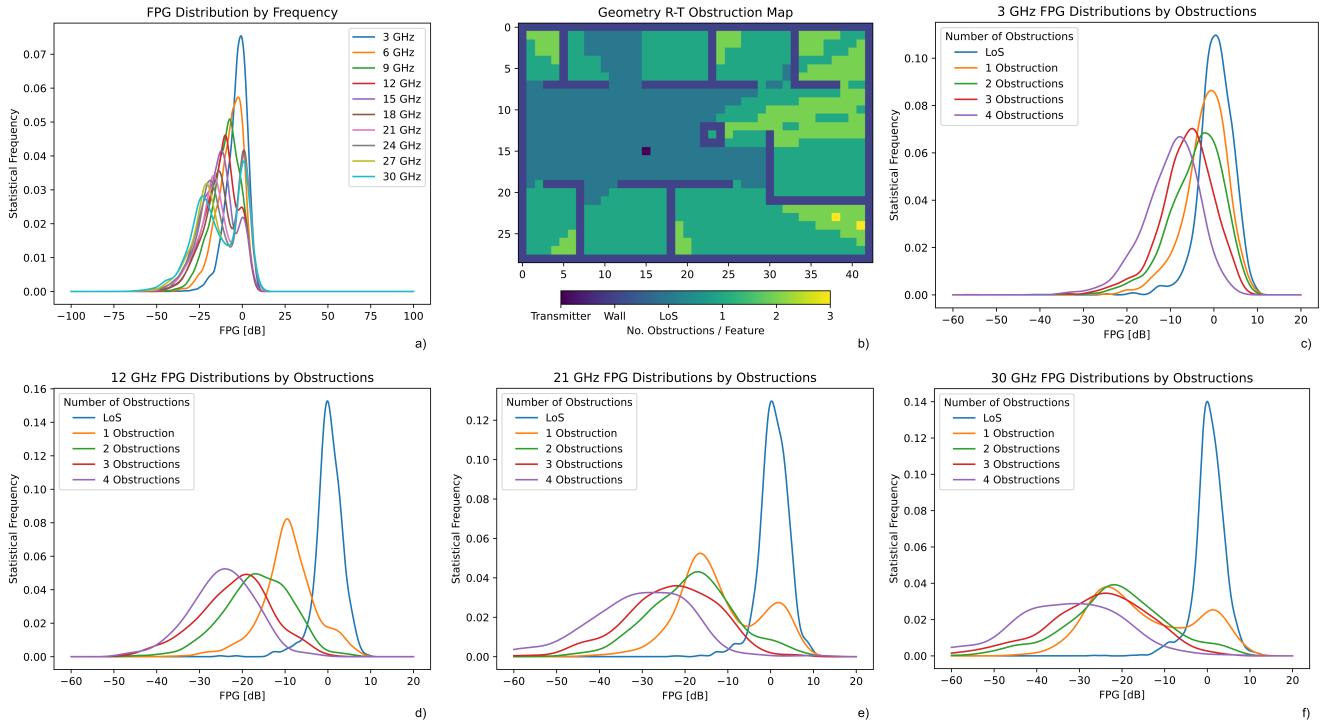


Fig. 11: An analysis of the general frequency case; we segmented the FPG values by number of obstructions between the receiver and transmitter (R-T). a) illustrates the bimodality experienced by the FPG distributions and the effect of frequency on the distribution. b) illustrates an example of R-T segmentation for our fixed geometry and a random transmitter position. c) - f) illustrate the distributions exhibited by FPG values in both the LoS and multi-obstruction nLoS cases for different frequencies.

slightly inferior in the general geometry cases. The results of the comparison are listed in Table III.

Experiment Case	LF-MAE [dB]	FPG-GAN MAE [dB]	RSS-GAN MAE [dB]	FPG-GAN Win Rate [%]
Position	12.144	2.490	2.530	75
Frequency	7.679	2.339	2.460	99
Geometry	26.153	2.280	2.284	45

TABLE III: A summary of the aggregate results of the FPG-GAN simulations compared to RSS-GAN. We compare the MAEs of both models and the win rate of FPG-GAN over RSS-GAN.

### B. Exploring Frequency Bimodality

As we noted in Section IV, the distribution of FPG values in the general frequency case follows a bimodal distribution, unlike the quasi-Gaussian distributions seen in the other two test cases – this effect was illustrated in Figure 10 b). In Figure 11 a) we show the decomposition of the distribution into frequency-specific distributions. Under this decomposition, we observe that this effect is most pronounced for frequencies above 9 GHz. To this effect, as the frequency increases, the observed distributions become more left-skewed.

In Figure 11 c)-f), we demonstrate that the FPG bimodality is the result of a mix of Gaussians. By grouping the receiver

points by the number of obstructions along the receiver-transmitter (R-T) path, at select frequencies, we deduce the following properties:

- 1) The distributions of the LoS components appear unchanged across the entire range of frequencies tested – they have near-zero means and narrower variances when compared to the other curves;
- 2) the greater the transmission frequency, the greater the leftward shift and variance of nLoS components;
- 3) the greater the number of obstructions, the greater the leftward shift and variance of nLoS components;
- 4) and the nLoS distributions are quasi-Gaussian in nature, except for the single-obstruction case which appears as a superposition of the LoS and higher-order obstructed cases.

As expected, we observe that rays which encounter more obstacles experience more attenuation, and that the rate of such attenuation is increased at higher frequencies. The fact that our FPG-GAN model can accurately capture these relations across a wide range of microwave frequencies makes it a promising candidate for modelling fifth- and sixth-generation (5G/6G) networks, which are prone to fading due to their high frequencies.

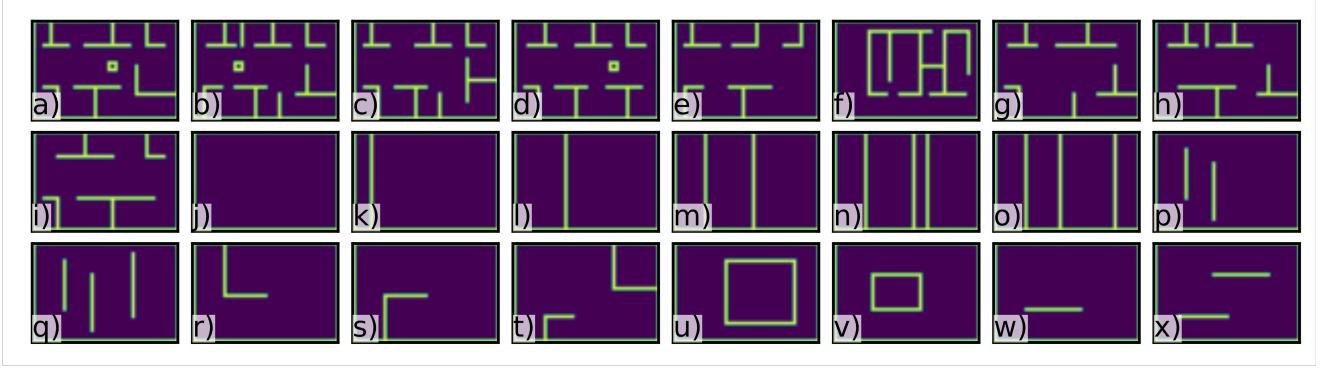


Fig. 12: The geometries used for training the GAN model. a)-f) belong to the *complex* geometry class. g)-x) belong to the *simple* geometry class.

## VI. CONCLUSION

To conclude, we presented a GAN-based wireless indoor propagation model which uses physics-based stochastic modeling to enhance low-fidelity RT predictions of RSS. Our model achieves this by considering the RSS to be made up of two components: the trivially modeled free-space component and the more complex component associated with multipath propagation. Our model demonstrates an ability to accurately predict the latter component concerning three main areas of generalizability: transmitter positions, transmission frequencies, and environmental geometries. By decomposing the total RSS into a free-space component and a faded component, we demonstrate an increased affinity for GANs to learn the geometry dependence of multi-path fading.

In each of the generalizations we tried, the model we present shows promising results, with error magnitudes being lower than in classical approaches that attempt to predict the total RSS. Yielding significant savings in terms of computational cost over deterministic high-fidelity RT, our model can be used to develop competitive alternatives to measurement-based channel characterization, with increased speed and accuracy.

## APPENDIX I TRAINING GEOMETRIES

The geometries used for the training of our GAN model are depicted in Figure 12. These geometries, along with those in our testing set, were produced by Seretis et. al. [2].

## REFERENCES

- [1] A. F. Molisch, *Wireless Communications*, ch. 2, pp. 27–36. John Wiley & Sons, Ltd, 2nd ed., 2011.
- [2] A. Seretis and C. D. Sarris, “Toward physics-based generalizable convolutional neural network models for indoor propagation,” *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 6, pp. 4112–4126, 2022.
- [3] D. Franci, S. Coltellacci, E. Grillo, S. Pavoncello, T. Aureli, R. Cintoli, and M. D. Migliore, “Experimental procedure for fifth generation (5g) electromagnetic field (emf) measurement and maximum power extrapolation for human exposure assessment,” *Environments*, vol. 7, no. 3, 2020.
- [4] A. F. Molisch, *Wireless Communications*, ch. 7, pp. 125–143. John Wiley & Sons, Ltd, 2nd ed., 2011.
- [5] Z. Yun and M. F. Iskander, “Ray tracing for radio propagation modeling: Principles and applications,” *IEEE Access*, vol. 3, pp. 1089–1100, 2015.
- [6] S.-H. Chen and S.-K. Jeng, “Sbr image approach for radio wave propagation in tunnels with and without traffic,” *IEEE Transactions on Vehicular Technology*, vol. 45, no. 3, pp. 570–578, 1996.
- [7] M. Lecci, P. Testolina, M. Polese, M. Giordani, and M. Zorzi, “Accuracy versus complexity for mmwave ray-tracing: A full stack perspective,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 7826–7841, 2021.
- [8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.
- [9] A. Seretis, T. Hashimoto, and C. D. Sarris, “A generative adversarial network approach for indoor propagation modeling with ray-tracing,” in *2021 IEEE International Symposium on Antennas and Propagation and USNC-USRI Radio Science Meeting (APS/URSI)*, pp. 657–658, 2021.
- [10] L. Azpilicueta, M. Rawat, K. Rawat, F. M. Ghannouchi, and F. Falcone, “A ray launching-neural network approach for radio wave propagation analysis in complex indoor environments,” *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 5, pp. 2777–2786, 2014.
- [11] D. Cui, G. Yang, S. Ji, S. Luo, A. Seretis, and C. D. Sarris, “Physics-informed machine learning models for indoor wi-fi access point placement,” in *2021 IEEE International Symposium on Antennas and Propagation and USNC-USRI Radio Science Meeting (APS/URSI)*, pp. 227–228, Dec 2021.
- [12] J. S. Seybold, *Introduction to RF Propagation*, ch. 9, pp. 208–217. John Wiley & Sons, Ltd, 2005.
- [13] P. Angueira and J. A. Romo, *Microwave Line of Sight Link Engineering*, ch. 7, pp. 227–270. John Wiley & Sons, Ltd, 2012.
- [14] A. Aragón-Zavala, *Indoor Wireless Communications*, ch. 5, pp. 113–166. John Wiley & Sons, Ltd, 2017.
- [15] S. Büyükkörok, M. Vural, and G. K. Kurt, “Lognormal mixture shadowing,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4386–4398, 2015.
- [16] S. Bakirtzis, T. Hashimoto, and C. D. Sarris, “FDTD-based diffuse scattering and transmission models for ray tracing of millimeter-wave communication systems,” *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 6, pp. 3389–3398, 2021.
- [17] A. Khanna, N. D. Londhe, S. Gupta, and A. Semwal, “A deep residual u-net convolutional neural network for automated lung segmentation in computed tomography images,” *Biocybernetics and Biomedical Engineering*, vol. 40, no. 3, pp. 1314–1327, 2020.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [19] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015.

- [20] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-realistic single image super-resolution using a generative adversarial network,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114, 2017.
- [21] S. Jenni and P. Favaro, “On stabilizing generative adversarial training with noise,” 2019.