# Millimeter-Wave Statistical Channel Modeling for Indoor Factories Using Generative Nets - Midterm Update

Shihao Ju and William Xia
*NYU Tandon School of Engineering, NYU WIRELESS, Brooklyn, NY, 11201*
{shao, william.xia}@nyu.edu

## I. Problem Statement

### A. Background

The vast amounts of bandwidth (over tens of GHz) offered by the Millimeter-wave (mmWave) and Terahertz (THz) spectrum [1] have the potential to support rapidly increasing mobile traffic demand throughout the world, which is expected to reach 77 exabytes per month by 2022 [2]. Advancements in technologies that operate in these frequencies (e.g. massive multiple input output (MIMO)) aims to support such stringent demands, but introduce more difficulties and challenges. Conventional channel modeling approaches may not accurately characterize the wireless channel given the increased number of complex modeling components. Thus, deep learning is considered to be a potential solution to this problem from a data-driven perspective. This project aims to employ a generative neural network to predict statistical channel information by using the simulated channels generated from a ray tracer as the training data.

### B. Conventional Channel Modeling

A received signal is commonly viewed as a superposition of multiple replicas of the transmitted signal with different delays and angles for any wireless propagation channel. Conventional channel modeling approaches can be classified into two types: deterministic approach and statistical approach. Deterministic approach describes the channel by modeling each propagation paths through free space propagation, reflection, diffraction, and scattering. Given a site-specific environment, the dimensions and building materials are required to generate propagation paths. Such deterministic approach is also well known as ray tracing, and high computation complexity and demanding requirements on the details of the environment are major concerns.

Instead, the statistical modeling approach considers the channel as a random variable, which usually cannot accurately recreate the channel conditions in a certain environment, but is capable of realizing the probabilistic distribution of a type of environment. A widely used cluster-based statistical channel model is given by

$$\begin{aligned} h(t, \overrightarrow{\Theta}, \overrightarrow{\Phi}) = &\sum_{n=1}^{N} \sum_{m=1}^{M_n} a_{m,n} e^{j\varphi_{m,n}} \cdot \delta(t - \tau_{m,n}) \\ &\cdot \delta(\overrightarrow{\Theta} - \overrightarrow{\Theta_{m,n}}) \cdot \delta(\overrightarrow{\Phi} - \overrightarrow{\Phi_{m,n}}), \end{aligned} \tag{1}$$

where $t$ is the absolute propagation time, $\overrightarrow{\Theta} = (\phi_{\text{AOD}}, \theta_{\text{ZOD}})$ is the angle of departure (AOD) vector, and $\overrightarrow{\Phi} = (\phi_{\text{AOA}}, \theta_{\text{ZOA}})$ is the angle of arrival (AOA) vector. $N$ and $M_n$ denote the number of clusters and the number of subpaths within each cluster, respectively. For the $m$th subpath in the $n$th cluster, $a_{m,n}$, $\varphi_{m,n}$, $\tau_{m,n}$, $\overrightarrow{\Theta_{m,n}}$, and $\overrightarrow{\Phi_{m,n}}$ represent the magnitude, phase, absolute time delay, AOD vector and AOA vector, respectively. As shown above, each channel parameter is fitted by some distribution, but the correlations among these channel parameters may not be accurately reserved.

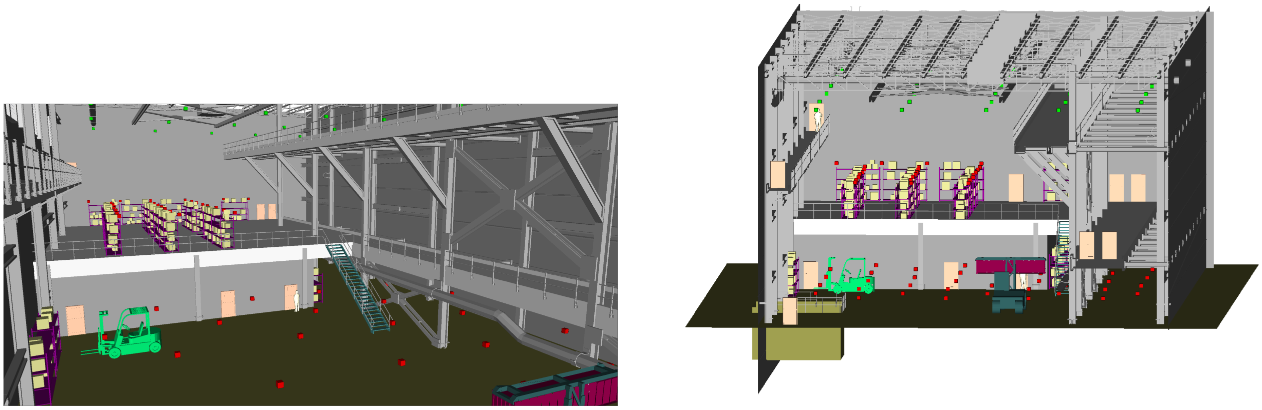
### C. Generative Channel Modeling

As the advent of deep learning, people start thinking about how deep learning techniques can be leveraged in wireless communications. Our idea is to train neural networks so that the neural networks can generate life-like channel information given a set of input parameters such as frequency, environment type, and separation distance.

There are a few attempts to apply generative neural networks to create wireless channels in various settings. A generative adversarial network(GAN)-based wireless channel modeling framework was proposed, where a vanilla GAN was used to generate additive Gaussian white noise (AWGN) channel without any domain-specific knowledge or technical expertise [3]. An experienced deep reinforcement learning (DRL) framework based on GANs was proposed to provide model-free resource allocation for ultra reliable low latency communication [4]. A variational GAN was trained to approximate wireless channel responses to more accurately reflect the probability distribution functions (PDFs) of stochastic channel behaviors [5]. In addition, conditional variational autoencoder was used as an alternative generative model to generate pathloss and angular information of multipath components (MPCs) [6].

## II. Data Acquisition and Processing

We produce our own data set for this project by using the a commercial raytracer: Wireless Insite Suite from Remcom [7]. The first environment that we attempt to train a model to generate samples for is the 'Factory Warehouse' scenario. This is shown in Fig. 1, where the receivers are denoted with red nodes, and the transmitters are denoted with green nodes. Note that complex environmental factors such as reflection



(a) Inside View

(b) Outside View

Fig. 1: Factory warehouse environment including transmitters and receivers.

coefficients and material permittivities of various objects like boxes, doors, guard rails, walls, windows, etc. are all considered in the simulator.

The parameters of the simulation performed are given in Table I.

TABLE I: Simulator configuration

| Parameter | Value |
|---|---|
| Number of TX | 35 |
| Number of RX | 67 |
| Max Number of Paths | 20 |
| Max Number of Reflections | 6 |
| Max Number of Diffractions | 1 |
| Carrier Frequency | 28 GHz |

The output of the ray-tracing simulation contains many channel statistics such as the the number of paths per transmission link, as well as the received power ($P_r$), propagation delay ($\tau$), AOA ($\phi_{AOA}, \theta_{ZOA}$), and AOD ($\phi_{AOD}, \theta_{ZOD}$) of each path. In order to process all the output data, a script was written to parse the data using the following format:

- Data structure $x$ is of size $n_{\text{samp}} \times 308$
- Each row of the $x$ represents the statistics of a link between a single TX-RX pair
- The first three columns give the TX location in cartesian coordinates, while the next three similarly give the RX location. This is followed by the number of rays in the link, the link state (LOS or NLOS), and the number of paths.
- The following 300 columns come from the 20 possible paths each of which contain the following 12 parameters: received power, path propagation delay, azimuth & elevation of departure, azimuth & elevation of arrival, cartesian coordinates of the first interaction after the transmitter, and finally the coordinates of the interactions before the receiver.

The data was processed in this way to handle the fact that any link can have a variable number of paths. For example, a link which has only 2 paths will have $8 + 2 \times 12 = 32$ columns of usable data, with all subsequent columns in that row being empty.

The parsed dataset was then further processed and put into a tensorflow dataset. Specifically, the final dataset has 124 columns. The first four columns represent the 3-D distance between the transmitter and receiver position ($d_{\text{3D}}$), the log of the 3-D distance ($10 \log_{10}$ 3D, the vertical distance ($d_z$), and the visibility condition (LOS or NLOS). The following 120 columns are composed by the six attributions of each of the 20 paths (i.e., $P_r, \tau, \phi_{AOA}, \theta_{ZOA}, \phi_{AOD}, \theta_{ZOD}$). These sample links all have these parameters which naturally have their own respective distributions. Thus the goal of this project is to use this data set to train a generative model that is capable of reproducing link data that follows the same distribution as the actual links that are found in the given environment.

## III. Network Structure and Implementation

### A. Generative Adversarial Networks

Generative models have been widely discussed in the machine learning community. Unlike the conventional discriminative models which predict a label given a set of features, the generative models are designed to create samples that mimic the features of actual data samples. GAN was first proposed by Goodfellow *et. al* in 2014 [8], then becomes one of the most interesting ideas and hottest research topics in the generative modeling field over the past few years. GAN differentiates itself from other generative models by exploiting a neural net as a expert to tell whether the input sample (either generated sample from the generator net or the real sample from real-world data) is fake or real. Such a expert is called discriminator, which needs to be trained along with the generator.

Simply put, the GAN consists of two neural nets, one generator and one discriminator, as shown in Fig. 2. By training these two neural nets with two different objective functions, the generator is trained to fool the discriminator by generating realistic samples, and the discriminator is trained to distinguish between the generated fake samples and the real samples. Such design realizes a adversarial setting. Precisely,

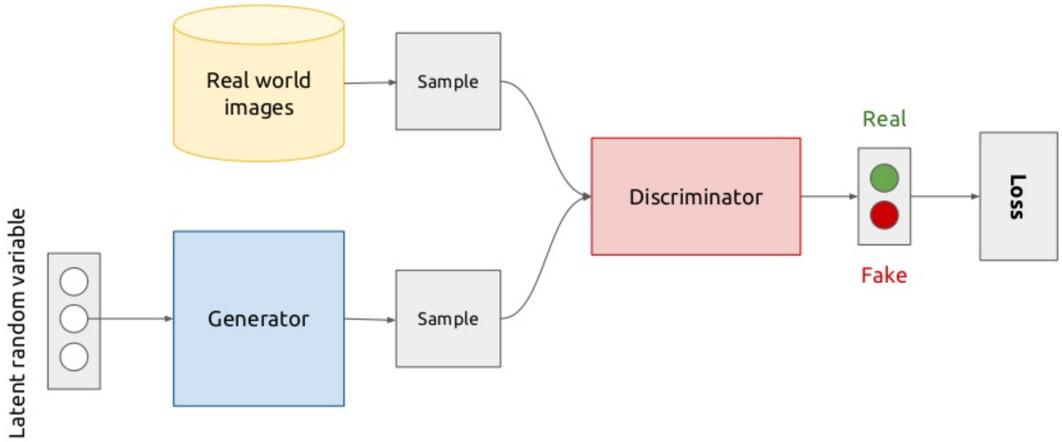



Fig. 2: GAN consists of two neural nets, one generator and one discriminator [9].

the generator and the discriminator are expected to reach Nash equilibrium after the training phase.

Assume the real-world data $x$ follows a probabilistic distribution $p_{\text{data}}(x)$, and the input to the generator $z$ is white noise with a prior distribution $p_z(z)$ (e.g., standard normal distribution). The generator maps the input noise vector to the data space $G(z;\theta_g)$, which $\theta_g$ parameterizes the generator neural net. On the other hand, the discriminator maps the input sample (data space) to a scalar $D(x)$ which represents the probability of assigning the correct label. Thus, the objective function of the discriminator is to maximize the probability of assigning the correct label to the real-world samples and to minimize the probability of assigning the correct label to the generated fake samples. On the contrary, the objective function of the generator is to maximize the probability of the discriminator assigning the correct label to its generated samples. Such settings forms a two-player minimax game with value function V(D,G) [8]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{2}$$

The hyperparameter settings of training the generator and discriminator are given in Table. II. The size of the input noise vector (latent variables) is set to 20, and the size of each input sample to the discriminator is 124 (as explained in Section II). Both the generator and the discriminator have two dense hidden layers with the LeakyReLU activation function. During each training iteration, the discriminator is updated based on the gradient of the discriminator loss function calculated by setting the labels of real-world samples as one and the labels of generated samples as zero; the generator is updated based on the gradient of the generator loss function by setting the labels of the generated samples as zero. After the GAN is trained, 1000 samples are created from the generator and evaluated in the Section IV.

TABLE II: HYPERPARAMETERS OF THE GENERATOR AND DISCRIMINATOR

| Hyperparameters | Generator | Discriminator |
|---|---|---|
| Number of inputs | 20 | 124 |
| Hidden unit | [32, 64] | [64, 32] |
| Number of outputs | 124 | 1 |
| Optimizer | Adam | Adam |
| Learning rate | 1e-4 | 1e-4 |
| Epochs | 100 | 100 |
| Batch size | 256 | 256 |
| Loss function | Binary cross-entropy | Binary cross-entropy |

## IV. PRELIMINARY RESULTS

Path loss is the most important performance metric for wireless communications. A good estimation and prediction of the path loss facilitates the design of cell coverage, handover schemes, and system throughput, etc. We use the close-in free space reference distance (CI) path loss model with 1 m reference distance [10]. $\text{PL}^{\text{CI}}$ represents the path loss in dB scale, which is a function of distance and frequency:

$$\text{PL}^{\text{CI}}(f,d)[\text{dB}] = \text{FSPL}(f,d_0) + 10n\log_{10}\left(\frac{d}{d_0}\right) + \chi_\sigma, \quad \text{for } d \geq d_0, \quad \text{where } d_0 = 1\text{m} \tag{3}$$

where $n$ denotes the path loss exponent (PLE), and $\chi_\sigma$ is the shadow fading (SF) that is commonly modeled as a lognormal random variable with zero mean and $\sigma$ standard deviation in dB. $d$ is the 3-D T-R separation distance. $d_0$ is the reference distance, and $\text{FSPL}(f,d_0) = 20\log_{10}(4\pi d_0 f/c)$. The CI path loss model uses the FSPL at $d_0 = 1$ m as an anchor point and fits the measured path loss data with a straight line controlled by a single parameter $n$ (PLE) obtained via the minimum mean square error (MMSE) method.

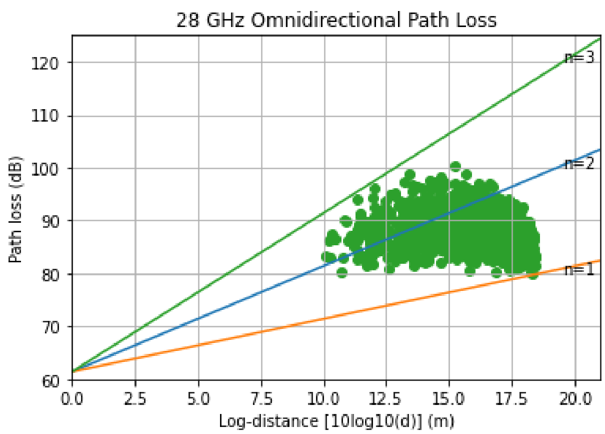



Fig. 3: The omnidirectional path loss created from the generator after 100-epoch training.

The PLE ($n$) of free space propagation is 2. The omnidirectional received power in the LOS scenario includes not only the LOS free space propagation path, but many reflected and scattered paths. Thus, the measured PLE of omnidirectional channels is usually less than 2. The omnidirectional PLE for the NLOS scenario usually ranges from 2 to 3. From Fig. 3, the created omnidirectional path losses at various distances and locations range between 1 and 3, which agrees with our expectation.

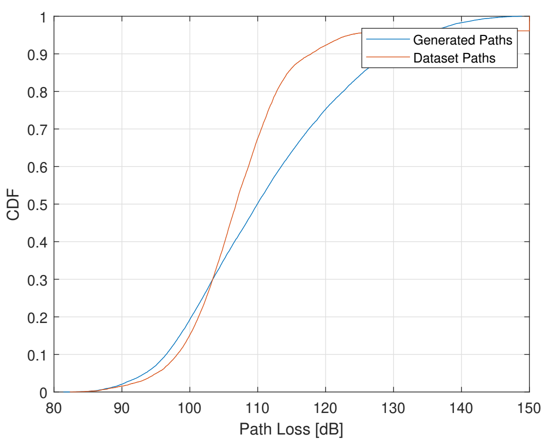



Fig. 4: The cumulative distribution of the path loss for paths across NLOS links

Next we plot the cumulative probability distribution (CDF) of the path losses from the NLOS links of both the test data, and of the link generated from our model. We can see that currently, the model generates links that follow the true distribution for paths that are at approximately $105\,\mathrm{dB}$ and below. However, for the remainder of the NLOS links, the GAN is overly pessimistic and generates paths that have path losses that are worse than the original data.

## V. Ongoing Works

- Continue the literature survey on the latest research progress on generative models. A brief literature review mainly on generative channel modeling is given in the mid-term update due to the space limit. A detailed literature review on general generative models will be included in the final report.
- Evaluate the GAN with respect to several other key channel parameters such as RMS delay spread and RMS angular spread. Preprocessing methods may be needed to improve the performance of the current GAN implementation.
- Experiment conditional GAN (CGAN) [11]. CGAN is to train a GAN with a conditioner or some extra information. For example, given the link state as LOS, CGAN can generate channel response for the LOS scenario.

## References


[1] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond," *IEEE Access*, vol. 7, pp. 78 729–78 757, 2019.

[2] CISCO, "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022," 2019.

[3] Y. Yang, Y. Li, W. Zhang, F. Qin, P. Zhu, and C. Wang, "Generative-adversarial-network-based wireless channel modeling: Challenges and opportunities," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 22–27, 2019.

[4] A. T. Z. Kasgari, W. Saad, M. Mozaffari, and H. V. Poor, "Experienced deep reinforcement learning with generative adversarial networks (gans) for model-free ultra reliable low latency communication," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 884–899, 2021.

[5] T. J. O'Shea, T. Roy, and N. West, "Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, 2019, pp. 681–686.

[6] W. Xia, S. Rangan, M. Mezzavillla, A. Lozano, G. Geraci, V. Semkin, and G. Loianno, "Generative neural network channel modeling for millimeter-wave uav communication," in *2020 GLOBECOM - IEEE Global Communications Conference Workshop*, 2020, pp. 1–6.

[7] "Remcom," available on-line at `https://www.remcom.com/`.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[9] K. McGuinness. (2017) Generative models and adversarial training (d3l4 2017 upc deep learning for computer vision). [Online]. Available: slideshare.net/xavigiro/generative-models-and-adversarial-training-d3l4-2017-upc-deep-learning-for-computer-vision

[10] S. Ju, Y. Xing, O. Kanhere, and T. S. Rappaport, "3-D statistical indoor channel model for millimeter-wave and sub-Terahertz bands," in *2020 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2020, pp. 1–7.

[11] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.