Physics-based Generative Models for Geometrically Consistent and Interpretable Wireless Channel Synthesis

Satyavrat Wagle[1], **Akshay Malhotra**[2], **Shahab Hamidi-Rad**[2], **Aditya Sant**[2], **David J. Love**[1] and **Christopher G. Brinton**[1]
[1]Elmore Family School of Electrical and Computer Engineering, Purdue University
[2]InterDigital Communications
wagles@purdue.edu, firstname.lastname@interdigital.com, {djlove,cgb}@purdue.edu

Abstract

In recent years, machine learning (ML) methods have become increasingly popular in wireless communication systems for several applications. A critical bottleneck for designing ML systems for wireless communications is the availability of realistic wireless channel datasets, which are extremely resource-intensive to produce. To this end, the generation of realistic wireless channels plays a key role in the subsequent design of effective ML algorithms for wireless communication systems. Generative models have been proposed to synthesize channel matrices, but outputs produced by such methods may not correspond to geometrically viable channels and do not provide any insight into the scenario being generated. In this work, we aim to address both these issues by integrating established parametric, physics-based geometric channel (PPGC) modeling frameworks with generative methods to produce realistic channel matrices with interpretable representations in the parameter domain. We show that generative models converge to prohibitively suboptimal stationary points when learning the underlying prior directly over the parameters due to the non-convex PPGC model. To address this limitation, we propose a linearized reformulation of the problem to ensure smooth gradient flow during generative model training, while also providing insights into the underlying physical environment. We evaluate our model against prior baselines by comparing the generated, scenario-specific samples in terms of the 2-Wasserstein distance and through its utility when used for downstream compression tasks.

1 Introduction

The use of machine learning (ML) for applications in wireless communication has seen extensive interest in the past few years. At the physical layer (PHY) of wireless communication systems, ML research has predominantly focused on two main objectives: estimating and mitigating distortions in electromagnetic signals during over-the-air (OTA) transmission (such as channel estimation, channel compression, equalization, and beamforming), [Liang *et al.*2020, Soltani *et al.*2019, Huang *et al.*2019, Mao *et al.*2018, Sant *et al.*2022] and addressing noise and non-linearities at transmitting or receiving antennas [Drakshayini and Kounte2022, Sant and Rao2024, Nguyen *et al.*2021]. For practical PHY layer deployments, the ML pipeline requires a substantial amount of OTA wireless channel data to effectively train the models. However, the process of manually collecting, cleaning and labeling wireless data from the real world is often complex and expensive both in terms of resources and time. Past data measurement and labeling campaigns have taken multiple months to capture a handful of fully characterized data points for a single scenario [Ju and Rappaport2023, Ju *et al.*2022, Kumar *et al.*2024].

Capitalizing on the recent advances in AI, generative models have been proposed to mitigate this problem by artificially synthesizing wireless data [Xiao *et al.*2022], significantly reducing the effort required to create wireless channel datasets for the aforementioned applications. However, unlike common modalities of data that we typically encounter, such as image, text, audio, etc. which are directly human-interpretable, the wireless channel data is a tensor of complex numbers and is *not human-interpretable or easily visualized*. Combined with the inherently stochastic nature of generative models, this poses two major challenges around effectively testing and using generative channel models. Firstly, the outputs of generative models may not correspond to valid channels. Here, the validity of channel data implies that the wireless channel can be represented as a multipath geometric model representing the multiple paths the transmitted signal takes, before arriving at the receiver [Forenza *et al.*2007, Meijerink and Molisch2014]. While the generative models are trained to generate data points statistically similar to the training set, owing to the stochasticity of the model, the synthesized outputs may not correspond to a geometrically valid set of interactions or multipaths. Secondly, it is hard to gain any insights about the physical parameters associated with the signal propagation or any information about the environment or scenario being considered (e.g. angles associated with paths, gains of paths, line-of-sight transmission or non-line of sight, etc.) from generated data samples.

This work primarily focuses on the design of generative

This work was presented in part at the 2025 ICLR Workshop DeLTa [Wagle *et al.*2025].

models for synthesizing millimeter wave (mmWave) channels, which forms the backbone of next-generation wireless communication and IoT systems [Kong *et al.*2017, Qamar *et al.*2019]. The proposed method to generate mmWave channels overcomes the limitations of existing approaches by incorporating a verified PPGC model into the generative pipeline. As the PPGC model parametrizes the channel generation, our generative process learns the joint distribution of the underlying parameters responsible for channel generation.

This mitigates both the above-mentioned issues, as by incorporating a verified PPGC model in the pipeline, the outputs of our framework are guaranteed to be valid channels, and as our model generates the parameters associated with the channels, we can extract insights related to the environment in which the channels were recorded, making the channels generated by our system more interpretable.

We further discuss the challenges associated with training the generative model with the PPGC model in the loop due to the non-convex loss landscape induced by the PPGC model. We further propose a linearized reformulation of the PPGC model, which mitigates these issues, and enables the integration of the PPGC model in the generative pipeline.

1.1 Summary of Contributions

The contributions of our paper are as follows.

- We propose a generative ML framework which leverages a parametric, physics-based channel (PPGC) model to produce realistic channel data that belongs to the distribution of interest (Sec. 3).
- We explore the challenges associated with the training of the proposed generative framework arising from the non-convexity of the PPGC model (Sec. 3.2).
- We develop a linearized relaxation of the PPGC model to mitigate the effect of this non-convexity (Sec. 3.3).
- We show that our method can accurately generate channel data as well as the parameter distributions associated with a given set of real data (Sec. 4).
- We evaluate our method against prior arts baselines, and experimentally show that our method is able to capture scenario-specific distributions more accurately (Sec. 4).

2 Related Work

Several works propose using a generative model to produce novel channel samples through the stochastic generative process. A generative adversarial network (GAN) based wireless channel modeling framework was first introduced in [Yang *et al.*2019]. [Xiao *et al.*2022] utilize a Wasserstein-GAN with Gradient Penalty (WGAN-GP) to synthesize novel channel matrices given a limited set of training data points and are evaluated by cross-validating between real and synthesized data. [Orekondy *et al.*2022] trained their model on multiple-input multiple-output (MIMO) data, with a discriminator explicitly designed to learn the spatial correlation across the channel data. In [Sengupta *et al.*2023], a diffusion based generative model has been adopted to circumvent the issue of mode collapse in GANs. They further evaluate the overlap of generated data with real data by calculating the approximate 2-Wasserstein distance between the power spectra. Works such as [Arvinte and Tamir2022] utilize a score-based generative model to generate channel matrices and further extend the framework for channel estimation in noisy environments. All of the above works utilize a generative model to directly produce channel matrices at the output, but they have no guarantees on the validity of the generated channel data, with limited interpretability.

A similar research direction involves using labeled datasets to predict parameters associated with the wireless channel. In [Xia *et al.*2022], the authors use the locations of UAVs to predict the link state and the channel parameters sequentially, using a conditional variational autoencoder (VAE) to capture complex statistical relationships within the data. Similar to the previous work, in [Hu *et al.*2022] the authors develop a GAN based model to generate new instances of channel parameters given the location of UAVs. The above works require datasets labeled with metadata relating to the environment, locations of the transmitter or receiver, and the entire set of channel parameters, and can not evaluate the channel matrices directly. In contrast, our method does not require labeled data and can learn channel parameters directly from the channel matrix.

3 System Model and Approach

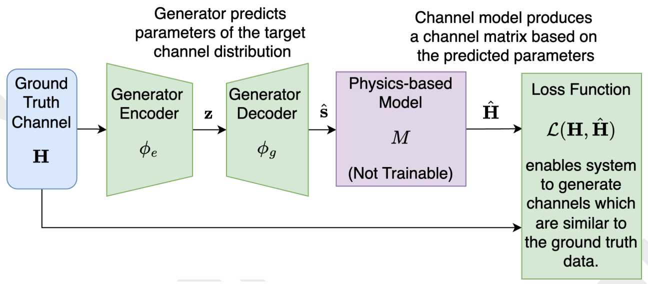
In this section, we describe our proposed system. We begin by describing the PPGC model that is used in the generative process, followed by the design of the generative model. We discuss the design choices made for the generative model. Finally, we describe the training and inference pipeline.

3.1 Channel Model

We consider a wireless communication system with N_t transmit and N_r receive antennas. The associated PPGC model defined by $\mathcal{M} : \mathbb{R}^{3P} \to \mathbb{R}^{2\times N_t\times N_r}$ [Alkhateeb *et al.*2014], maps a set of parameters $\mathbf{s} \in \mathbb{R}^{3P}$ to a matrix $\mathbf{H} \in \mathbb{C}^{N_t\times N_r}$, where P is the number of paths that a transmitted signal takes before being received at the receiver antennas. In mmWave channels, the total number of paths P is typically small in over-the-air transmission. The PPGC model considers the channel matrix $\mathbf{H}$ to be a superposition of the individual propagation matrices associated with each of the P paths.

$$\mathbf{H} = \mathcal{M}(\mathbf{s}) = \sum_{p=1}^{P} g_p \mathbf{a}_r(\theta_a^p)\mathbf{a}_t(\theta_d^p)^H. \tag{1}$$

Where, $\mathbf{s} = [g_p, \theta_a^p, \theta_d^p]_{p=1}^P$, g_p represents the propagation gain associated with the p-th path, $\mathbf{a}_t(\cdot), \mathbf{a}_r(\cdot)$ represent the steering vectors or the array response vectors on the transmit and receive antennas, and θ_d^p and θ_a^p represent the corresponding angle of departure and angle of arrival, both of which take values between $[-\pi, \pi]$ radians. To simplify the discussion, we consider a Uniform Linear Array (ULA) [Forenza *et al.*2007] and limit the discussion to the azimuth plane. Thus,


Generator predicts parameters of the target channel distribution
Channel model produces a channel matrix based on the predicted parameters
Ground Truth Channel
H
Generator Encoder
ϕ_e
z
Generator Decoder
ϕ_g
ŝ
Physics-based Model
M
(Not Trainable)
Ĥ
Loss Function
$\mathcal{L}(\mathbf{H}, \hat{\mathbf{H}})$
enables system to generate channels which are similar to the ground truth data.

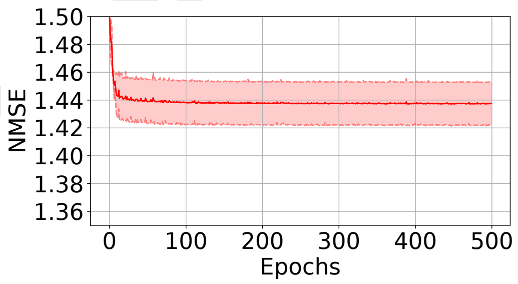


Figure 1: When integrating the PPGC model in a straightforward manner, the generator directly predicts the parameters $\hat{\mathbf{s}}$, which are then used by the PPGC model $\mathcal{M}$ to produce the predicted channel (Top). In such implementations, the generator cannot converge to suitable optima due to the non-convexity of the PPGC model, as observed in the training performance of the pipeline over multiple instances. (Bottom)

the array response vectors can be defined as:

$$\mathbf{a}_t(\theta_d^p) = \frac{1}{\sqrt{N_t}}[1, e^{ju\sin(\theta_d^p)}, \dots, e^{j(N_t-1)u\sin(\theta_d^p)}]^T, \quad (2)$$

$$\mathbf{a}_r(\theta_a^p) = \frac{1}{\sqrt{N_r}}[1, e^{ju\sin(\theta_a^p)}, \dots, e^{j(N_r-1)u\sin(\theta_a^p)}]^T, \quad (3)$$

where, $u = \frac{2\pi}{\lambda}d$, λ is the wavelength of the carrier and d is the distance between antenna elements.

A key characteristic of the PPGC model $\mathcal{M}$ is that, a distribution $q(\cdot)$ over the parameters $\mathbf{s}$ induces a distribution $q_M(\cdot)$ over the channel matrix $\mathbf{H}$, where the distribution $q(\cdot)$ is specific to the environment in which the model is deployed. Now, in order to incorporate the PPGC model into the generative pipeline, the generative model must learn the prior distribution over the parameters $\mathbf{s}$ instead of directly learning the prior distribution over the channel matrix $\mathbf{H}$. Thus, the objective of our system is as follows; given a set of channel matrices $\mathcal{D} = [\mathbf{H}_i]_{i=1}^N$, where $\mathbf{H}_i \sim q'(\cdot)$ and $q'(\cdot)$ is unknown, we train a generative model to output a probability distribution $q(\cdot)$ over the parameters $\mathbf{s}$ such that the induced distribution $q_M(\cdot)$ over the channel $\mathbf{H}$ is close to $q'(\cdot)$.

3.2 Generative Model to Predict Channel Statistics

For the generative model, we use the variational autoencoder (VAE) architecture [Kingma and Welling2014], as seen in Fig. 1. We use the generative model to produce the parameter vector $\mathbf{s}$ using a latent variable $\mathbf{z}$, which is then passed to the PPGC model $\mathcal{M}$ to produce a valid channel matrix. The VAE consists of a decoder network $g_{\phi_d}(\cdot)$, parametrized by ϕ_d, which is used to reconstruct the parameter vector $\mathbf{s}$ given the latent variable $\mathbf{z}$ as $g_{\phi_d}(\mathbf{z})$, and the encoder $f_{\phi_e}(\cdot)$, parametrized by ϕ_e, which approximates the posterior distribution of $\mathbf{z}$ given the channel matrix $\mathbf{H}$ using the variational posterior $f_{\phi_e}(\mathbf{H})$.

Generative model training

During the training phase, the encoder of the generative model takes a channel matrix $\mathbf{H}$ as input and samples a latent vector $\mathbf{z}$ from the posterior distribution as

$$\mathbf{z} \sim f_{\phi_e}(\mathbf{H}). \quad (4)$$

The decoder of the generative model then takes in the latent vector $\mathbf{z}$ as input and produces a parameter vector $\hat{\mathbf{s}}$ as

$$\hat{\mathbf{s}} = g_{\phi_d}(\mathbf{z}). \quad (5)$$

The predicted parameter vector is then passed to the model $\mathcal{M}$ to produce an output channel $\hat{\mathbf{H}}$ as follows

$$\hat{\mathbf{H}} = \mathcal{M}(\hat{\mathbf{s}}). \quad (6)$$

The system loss is a generalization of the evidence based lower bound (ELBO) [Kingma and Welling2014], given by

$$\mathcal{L} = ||\mathbf{H} - \hat{\mathbf{H}}||_2^2 + \alpha_D \cdot \mathsf{KL}(\mathbf{z}, \mathcal{N}(0, \mathbf{I})). \quad (7)$$

Here, the first term corresponds to the reconstruction or mean square error (MSE) loss between the input $\mathbf{H}$ and the predicted channel matrix $\hat{\mathbf{H}}$. This ensures that the outputs are similar to the inputs. The second term penalizes the Kullback-Leibler (KL) divergence [Kullback and Leibler1951] between the latent vector $\mathbf{z}$ and a simple, known distribution, in this case, the multivariate unit Gaussian distribution $\mathcal{N}(0, \mathbf{I})$, where $\mathbf{I}$ is the identity matrix of dimension Z. This encourages the distribution of the latent vectors to be similar to $\mathcal{N}(0, \mathbf{I})$.

Channel generation by incorporating the PPGC model proceeds in accordance with the standard VAE architecture. The loss function (7) enforces a latent space distribution similar to $\mathcal{N}(0, 1)$, with the decoder $g_{\theta_d}(\cdot)$ trained as the channel generator network. The input to the decoder is a random variable $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. The decoder generates a set of parameters $\hat{\mathbf{s}}$, corresponding to the physics-based parameters of a new channel. These parameters are passed through the PPGC model $\mathcal{M}$ to generate an interpretable and valid channel matrix. However, the direct use of the physics-based channel model, using (1) requires knowledge of the number of multipath components P. Further, gradient backpropagation using the PPGC model results in poor training performance, owing to the non-convexity of the model. This is further elucidated below.

Limitations of generative model training using PPGC

In order to understand the limitations of training a generative network directly in conjunction with the PPGC model (1), we analyze the role of the ULA manifold, given in (2). The presence of sinusoidal functions of the angular channel parameters, θ_d^p and θ_a^p, in creating the array response vector results in periodicities in the loss function landscape. The non-convexity arising from this periodicity is only exacerbated with the addition of multiple paths. However, these periodicities cannot be effectively approximated by the non-linear activation functions used in deep neural networks, leading to difficulties in training [Nair and Hinton2010, Dumoulin and

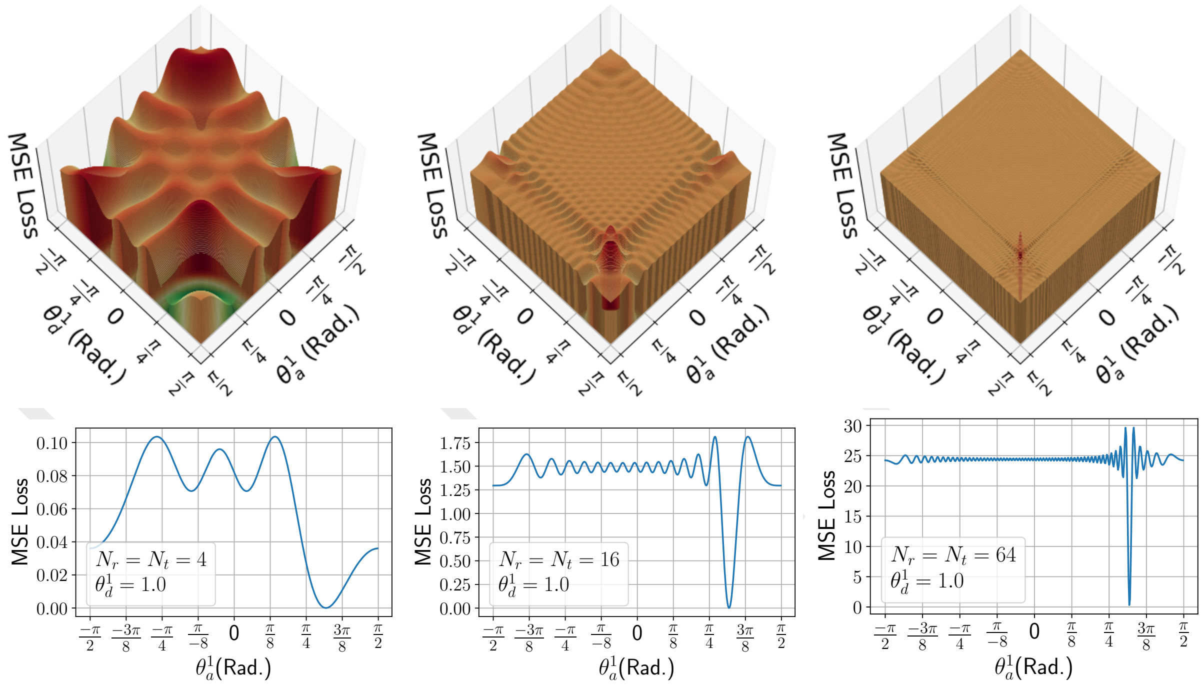


Figure 2: The loss surface as a function of (θ_a^1, θ_d^1) in reference to a channel matrix with $\theta_a^1 = \theta_d^1 = 1.0$ radians using a PPGC model $\mathcal{M}$ with $P = 1$ and $N_r = N_t = 4, 16, 64$ antennas respectively. The PPGC model $\mathcal{M}$ is extremely non-convex as a function of the parameters θ_a, θ_d because of periodicity arising from the formulation of the array response vectors. Avoiding the numerous local minima surrounding the global minima poses a significant challenge given the nature of activation functions. As the number of antennas N_r, N_t increases, the gradient flowing through the model diminishes significantly at locations further away from the minima, where the loss surface is effectively flat.

Visin2016]. As a result, depending on the location of the optimizer in the parameter space, the gradient may not flow during backpropagation, resulting in the optimizer converging at non-optimal points. Additionally, the convergence of the system to a suitable minimum is highly dependent on the number of antennas used in the PPGC model. This is illustrated in Fig. 2 where, as the number of antennas, N_t, N_r increase, the number of local minima also increase. Further, the optimality gap, the difference between the loss values at the local minima and the global minima, widens as the number of antennas increases, causing convergence to any local minima to significantly impact the overall loss, as observed in Fig. 1. Also, as observed from Fig. 2, the loss landscape flattens as parameter values move further from the global optimum, leading to smaller gradients. This diminishes the effectiveness of optimization processes and gradients in achieving the optimal parameter values, regardless of the optimization strategies and momentum employed.

3.3 Linearized Reformulation of the Physics Model

To overcome the challenges posed by the PPGC model, while maintaining the key underlying model features for channel generation, we relax the overall cost function by reformulating the channel generation model $\mathcal{M}$ by first discretizing the range of values that θ_a^p, θ_d^p can take and then expressing the channel generation process as a weighted average of the resulting antenna array responses vectors $\mathbf{a}_r(\theta_a^p)\mathbf{a}_t(\theta_d^p)^H$. Different from directly estimating the channel parameters, this method utilizes dictionary-based channel generation, where the loss function is now a linear function of the learnable dictionary weights. This is further explained next.

Let the range of angles θ_a^p, θ_d^p, given by $[\theta_{\min}, \theta_{\max}]$, be equally divided into R intervals of width $\Delta_\theta = (\theta_{\max} - \theta_{\min})/R$. We pre-compute the outer product between the array response vectors, $\mathbf{a}_r(\theta_a^p)\mathbf{a}_t(\theta_d^p)^H$, at the discretized angle values and store them in a dictionary $\mathbf{D}$. The dictionary $\mathbf{D}$ has a total of R^2 elements, and we refer to it as the array response dictionary across the remainder of this work. Each element of the dictionary, $\mathbf{D}_{i,j}\ \forall\, i, j \in \{1, R\}$, represents a $N_r \times N_t$ matrix, and is given by

$$\mathbf{D}_{i,j} = \mathbf{a}_r(\theta_i)\mathbf{a}_t(\theta_j)^H, \tag{8}$$

where $\left\{\theta_k := \theta_{\min} + k(\Delta_\theta) \middle|\ k \in \{i, j\}\right\}$. Thus, each element of the array response dictionary is the combination of antenna array responses at the transmitter and receiver for certain values of the angle of arrival and angle of departure. The angles associated with neighboring elements of the dictionary $\mathbf{D}_{i,j}, \mathbf{D}_{i+1,j}$ or $\mathbf{D}_{i,j}, \mathbf{D}_{i,j+1}$ differ by a value of Δ_θ for the angle of arrival or the angle of departure respectively. The relaxed PPGC model can now be expressed as,

$$\mathbf{H} = \sum_{i=1}^{R}\sum_{j=1}^{R} \mathbf{W}_{i,j}\mathbf{D}_{i,j}, \tag{9}$$

where, the channel generation is parametrized by the gain matrix $\mathbf{W} \in \mathbb{R}^{R\times R}$, instead of the parameters $\mathbf{s}$, and the

Algorithm 1 Generation of PPGC using Linearized Model

```
Training :
Given : Dataset of valid channels $\mathcal{D}$, VAE parametrized by $\phi_e, \phi_d$, Chosen range of angles $[\theta_{\min}, \theta_{\max}]$, Resolution $R$, Latent dimension $Z$.
1: Calculate the array response dictionary $\mathbf{D}$ using (8) for all $i \leq R, j \leq R$.
2: Sample a channel matrix $\mathbf{H}$ from the dataset $\mathcal{D}$.
3: Obtain gain matrix $\mathbf{W}$ from $\mathbf{H}$ using (4) and (10).
4: Obtain the predicted channel matrix $\hat{\mathbf{H}}$ from $\mathbf{W}$ using (11).
5: Calculate the loss using (12) and update $\phi_e, \phi_d$ =0
Generation :
Given : Sample vector $\tilde{\mathbf{z}} \in \mathbb{R}^Z$ from Multivariate unit Gaussian distribution $\mathcal{N}(0, \mathbf{I})$
1: Obtain gain matrix $\tilde{\mathbf{W}}$ from $\tilde{\mathbf{z}}$ using (13)
2: Obtain generated channel matrix $\tilde{\mathbf{H}}$ from $\tilde{\mathbf{W}}$ using (14) =0
```

channel is constructed by the element-wise product between $\mathbf{W}$ and $\mathbf{D}$.

By making these changes in the pipeline, we now model the output channel $\mathbf{H}$ as a linear function of the gain matrix $\mathbf{W}$, mitigating the issues arising from the non-convexity of the PPGC model $\mathcal{M}$ as seen in Sec. 3.2. Thus, we now use the generative model to predict $\mathbf{W}$ instead of $\mathbf{s}$. Now, as the total number of paths P is typically small, as mentioned in Sec. 3.1, only a few of the entries in the $\mathbf{W}$ are expected to be non-zero.

Remark 1 : It is to be noted that (1) and (9) both produce valid channel matrices. For a suitably high value of R, any channel $\mathbf{H}$ can be approximated by (9) using a suitable sparse $\mathbf{W}$ with P non-zero values. For such a $\mathbf{W}$, (9) is equivalent to (1), and each non-zero value $\mathbf{W}_{i,j}$ will correspond to the gain g_p of one of the P paths. Thus, for this linearized representation of the problem, we can interpret the non-zero values of the gain matrix $\mathbf{W}$ as the path gains, and the angles associated with the corresponding array dictionary vector $\mathbf{D}_{i,j}$ as the angle of arrival and departure of those paths.

3.4 Generative Model to Predict the Gain Matrix

In order to utilize the generative model to predict the gain matrix, we change the training pipeline described in Sec. 3.2 in the following ways.

The VAE decoder $g_{\phi_d} : \mathbb{R}^Z \to \mathbb{R}^{R\times R}$ now predicts the gain matrix $\mathbf{W}$, where R is the resolution of $\mathbf{W}$. During training, the decoder g_{ϕ_d} takes in the latent vector $\mathbf{z}$ as input and produces a gain matrix $\mathbf{W}$ as

$$\mathbf{W} = g_{\phi_d}(\mathbf{z}). \tag{10}$$

The gain matrix is then used to generate the predicted channel by multiplying it with the array response dictionary as

$$\hat{\mathbf{H}} = \sum_{i=1}^{R}\sum_{i=1}^{R} \mathbf{W}_{i,j}\mathbf{D}_{i,j}. \tag{11}$$

Now, we propose a modified system loss, that also accounts for the sparsity in the weight matrix $\mathbf{W}$. This is given by

$$\mathcal{L} = ||\mathbf{H} - \hat{\mathbf{H}}||_2^2 + \alpha_D \cdot \mathsf{KL}(\mathbf{z}, \mathcal{N}(0, \mathbf{I})) + \alpha_S \cdot ||\mathbf{W}||_1. \tag{12}$$

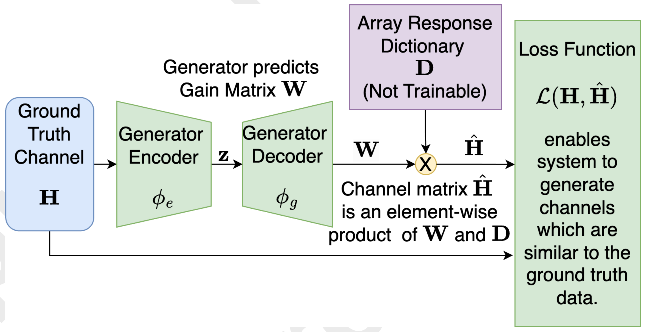

Ground Truth Channel
H
Generator Encoder
ϕ_e
z
Generator Decoder
ϕ_g
Generator predicts Gain Matrix W
Array Response Dictionary
D
(Not Trainable)
W
X
$\hat{\mathbf{H}}$
Channel matrix $\hat{\mathbf{H}}$ is an element-wise product of W and D
Loss Function
$\mathcal{L}(\mathbf{H}, \hat{\mathbf{H}})$
enables system to generate channels which are similar to the ground truth data.

Figure 3: We relax the PPGC model by defining a discretized array response dictionary $\mathbf{D}$ and using the generator to output the gain matrix $\mathbf{W}$. The elementwise multiplication of $\mathbf{W}$ and $\mathbf{D}$ mimics the PPGC model process. This relaxation allows the flow of gradient through the generator, enabling it to converge to more suitable optima.

Here, the first two terms are identical to the ones used in (7). The last term is the 1-norm of the gain matrix $\mathbf{W}$, which encourages $\mathbf{W}$ to be as sparse as possible, ensuring that $\mathbf{W}$ does not select unrealistic combinations of multipaths that minimize the MSE. Thus, the overall reformulation can be interpreted as a sparse dictionary learning problem, with an additional KL divergence term.

3.5 Inference and Sampling

During the inference phase, we only use the generative decoder g_{ϕ_d}. We sample a vector $\tilde{\mathbf{z}} \sim \mathcal{N}(0, \mathbf{I})$ and pass it through the decoder to produce a gain matrix $\tilde{\mathbf{W}}$

$$\tilde{\mathbf{W}} = g_{\phi_d}(\tilde{\mathbf{z}}) \tag{13}$$

$\tilde{\mathbf{W}}$ is then used to generate a synthetic channel, using the array response dictionary, as

$$\tilde{\mathbf{H}} = \sum_{i=1}^{R}\sum_{i=1}^{R} \tilde{\mathbf{W}}_{i,j}\mathbf{D}_{i,j}. \tag{14}$$

The generated channel $\tilde{\mathbf{H}}$ is guaranteed to be a valid channel matrix as the mapping from the parameter space to the channel space is done based on a verified PPGC model. $\tilde{\mathbf{H}}$ is also expected to be from the distribution of interest, as the second term in (12) ensures that the decoder of the generative model learns to map samples from the multivariate unit Gaussian distribution to a set of parameters that belong to the distribution of interest. A summary of the training and generative processes for the linearized representation is given in Algo. 1.

4 Experimental Results

In this section, we analyze the performance of our method on wireless datasets generated based on user-defined parameter distributions as well as those that correspond to real-life scenarios, and compare our method against prior art baselines. We show the efficacy of our method in capturing the underlying parameter distributions, its ability to accurately generate synthetic channels, as well as its effectiveness of the proposed generative pipeline for downstream channel compression tasks.

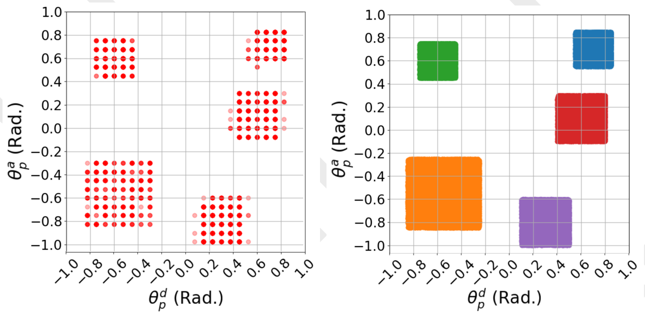


Figure 4: The distributions of angles of arrival and departure (θ_a^p, θ_d^p) captured by our method (Left) match the underlying distributions of the training dataset (Right). Each color corresponds to a distinct path.

For our PPGC model, we consider transmit and receive antennas $N_t = N_r = 16$. We consider six datasets. Five of the datasets are taken from the DeepMIMO framework [Alkhateeb2019] that utilizes 3D ray tracing for generating channel datasets in different settings. Datasets corresponding to the following scenarios are used; (i) Two base stations in an outdoor intersection of two streets with blocking and reflecting surfaces. Here the channels corresponding to base station 10 (BS10) and base station 11 (BS11) in the DeepMIMO framework have been considered; (ii) An indoor conference room, given by (Indoor); (iii) A section of downtown Boston, Massachusetts, USA, generated using the 5G model developed by RemCom [Remcom2025], given by (Boston) and (iv) A section of the Arizona State University campus in Tempe, Arizona, USA, given by (ASU). The last is a user-defined dataset of size $20,000$ produced by sampling known distributions of parameters and using those parameters with the PPGC model $\mathcal{M}$ to produce a dataset (User Defined). More information regarding the datasets is provided in Appendix Sec. B. All datasets are split $80/20$ for training and testing respectively.

We use a VAE as the generative model, (g_{ϕ_e}, g_{ϕ_d}), which is trained using the Adam optimizer with a learning rate of $1e^{-3}$. For more details regarding the model architecture, please refer to Appendix Sec. A. We compare our model against ChannelGAN (CGAN) [Xiao *et al.*2022], the DUNet diffusion model (DUNet) [Sengupta *et al.*2023] and a VAE version of CSINet [Wen *et al.*2018]. Models are trained for 300 epochs with a batch size of 256, resolution $R = 64$ and $z = 64$.

4.1 Prediction of Parameters

In this experiment, we evaluate the accuracy of our system in capturing the underlying distribution of parameters associated with a given dataset. We first define a distribution across the parameters $[g_p, \theta_p^a, \theta_p^d]_{p=1}^P$ and generate a dataset $\mathbf{D}$ of channel matrices by sampling parameters from this distribution and passing them through the model $\mathcal{M}$. We train our model on the dataset $\mathbf{D}$ and compare the distribution of predicted parameters $[\tilde{g}_p, \tilde{\theta}_p^a, \tilde{\theta}_p^d]_{p=1}^P$ with that of the true parameters $[g_p, \theta_p^a, \theta_p^d]_{p=1}^P$.

In Fig. 4, we observe that our method can accurately capture the distributions of the angles of arrival and departure for each path. The discretized array response dictionary results in grids of generated angles of arrival and departure, which align with the distribution of parameters used to generate the channels. This shows that our model training and parameter extraction methods can be used to determine the distributions of parameters of input channels without requiring labeled data.

4.2 Reconstruction of Channels

In Table 2, we analyze the ability of our method to capture the distribution of channel matrices compared to baseline methods. We generate 3000 synthetic channel matrices and compare the 2-Wasserstein distance [Panaretos and Zemel2019] and Maximum Mean Discrepancy [Dziugaite *et al.*2015] between the distribution of the generated channels and the true channels.

The channels generated by our method are closer to the distribution of true channels than those generated by the ChannelGAN baseline by up to $4\times$. This shows that our method can generate more realistic channel data as compared to baselines.

4.3 Cross Evaluation Between Distinct Datasets

In this experiment, we analyze the ability of our method to learn distinct channel distributions based on the cross-evaluation of models trained on different scenarios in the context of a downstream channel compression task. In practical wireless deployments, effective channel compression is crucial in mitigating feedback overheads that arise in systems that frequently exchange channel data to optimize communication efficiency [Wen *et al.*2018, Mashhadi *et al.*2020].

We consider two channel datasets R_{10}, R_{11} from the DeepMIMO Outdoor scenario, generated from base stations 10 and 11 respectively. We train an independent instance of the generative model on each dataset and generate synthetic datasets of size $20,000$, given by G_{10}, G_{11} respectively. We then train independent instances of the CSINet channel compression model [Wen *et al.*2018] on each set. We perform cross evaluation considering all pairwise combinations of training and testing datasets and calculate the test NMSE given in Table 1.

Now, a model trained on G_{10} should generalize well to R_{10}, and vice versa for a model trained on G_{11}. In Table 1, we observe that a compression model trained on data generated by our model follows the aforementioned rules, indicating that our method can capture the distinctions between two different datasets and generate distinct channel data samples.

4.4 Effect of Varying Size of Training Dataset

In this experiment, we train our model on datasets of varying sizes, choosing a subset of the original dataset of the appropriate size. We use the DeepMIMO dataset BS10 for this experiment, which consists of $16,000$ datapoints.

In Fig. 5(a), we observe that our method is able to provide a similar level of performance even when the size of the training data is reduced by up to $\sim 40\%$. This is because our model predicts distributions in the parameter space, which are less complex than the distributions in the channel space, even a small number of datapoints can capture the distribution related characteristics of the input channels. In a practical deployment, this translates to significant savings in terms of the resources deployed to acquire channel data.

4.5 Effect of Varying Resolution

In this experiment, we observe the joint effect of changing the resolution R of the gain matrix $\mathbf{W}$, and the number of antennas

Train	Testing R_{10}				Testing R_{11}				Testing G_{10}				Testing G_{11}			
	Ours	CGAN	DUNet	CVAE	Ours	CGAN	DUNet	CVAE	Ours	CGAN	DUNet	CVAE	Ours	CGAN	DUNet	CVAE
R_{10}	0.02	0.02	0.02	0.02	1.35	1.35	1.35	1.35	**0.06**	1.29	0.46	0.55	1.36	1.37	1.18	1.07
R_{11}	1.14	1.14	1.14	1.14	0.06	0.06	0.06	0.06	1.21	1.51	1.14	1.04	**0.25**	0.45	0.42	0.72
G_{10}	**0.05**	0.77	0.19	0.85	1.41	0.97	1.04	1.49	0.01	0.09	0.03	0.09	1.5	0.92	1.17	1.1
G_{11}	1.1	1.37	1.02	1.04	**0.14**	0.56	0.37	0.52	1.31	1.94	1.33	1.15	0.01	0.02	0.01	0.02

Table 1: NMSE loss for downstream compression tasks using different pairs of training and testing datasets. When compression models are trained on real data and evaluated on generated data (Rows 1,2) and vice versa (Rows 3,4), our method records lower NMSE for corresponding real-generated dataset pairs, indicating that the data generated by our method is more similar to the real channel data.

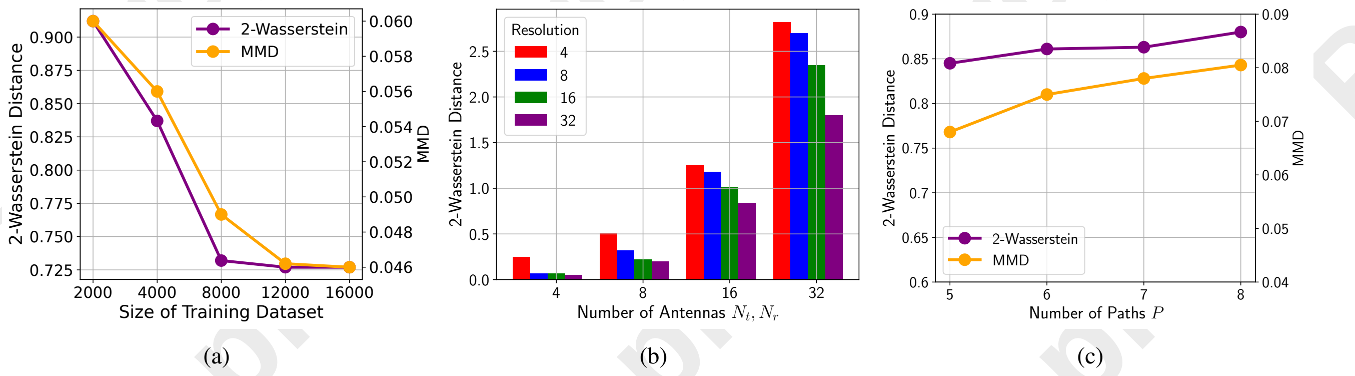


Figure 5: (a) Our method can generate samples with a high degree of fidelity in terms of 2-Wasserstein distance and MMD, even with a training dataset of around 50% of the size of the training dataset. (b) For a physics model $\mathcal{M}$, as the number of antennas (N_r, N_t) increase, an increase in the resolution (R) of the gain matrix $\mathbf{W}$ results in a higher degree of fidelity with the input data distribution in terms of the 2-Wasserstein distance, as the highest possible precision with which parameters can be estimated is dependent on the number of antennas. (c) The performance of our method is consistent across a varying number of paths P, as the generative process is independent of P, relying on the loss function to balance reconstruction fidelity and the identified number of paths, given by the number of non-zero values in the gain matrix $\mathbf{W}$.

	2-Wasserstein Distance				MMD			
Dataset	Ours	CGAN	DUNet	CVAE	Ours	CGAN	DUNet	CVAE
Boston	**0.475**	0.84	0.687	0.87	**0.013**	0.045	0.052	0.095
ASU	**0.283**	0.932	0.641	1.095	**0.012**	0.169	0.02	0.592
Indoor	**0.17**	0.339	0.529	1.802	**0.005**	0.003	0.02	0.375
BS10	**0.656**	1.632	0.857	1.837	**0.072**	0.317	0.108	0.414
BS11	**0.267**	0.882	0.463	1.12	**0.016**	0.086	0.033	0.132

Table 2: The distribution of channels modelled by our method is more similar to the real distribution compared to baselines in terms of 2-Wasserstein distance and Maximum Mean Discrepancy (MMD).

N_r, N_t used by the physics model $\mathcal{M}$. For this experiment, we use the DeepMIMO dataset for base station 10.

In Fig. 5(b), we observe that for a model with fewer antennas ($N_r = N_t = 4/8$), an increase in resolution R offers diminishing improvements. This is because the highest level of granularity at which parameters can be estimated is dependent on the number of antennas, with further increase in resolution resulting in little in performance. However, for a model with more antennas ($N_r = N_t = 16/32$), the added dimensionality of the channel matrices allows the model to identify angles of arrival and departure, $(\theta_a^p, \theta_d^p)_{p=1}^P$ with greater precision. Thus, in such cases, increasing the resolution of the weight matrix $\mathbf{W}$ improves the performance of the model considerably.

4.6 Effect of Increasing Number of Paths

In this experiment, we observe the effect of increasing the number of paths P. We consider the user defined dataset with additional paths as follows. Path 6 is sampled from $\theta_p^a \sim \mathcal{U}(0.4, 0.8)/\theta_p^a \sim \mathcal{U}(0.1, 0.3)$, Path 7 from $\theta_p^a \sim \mathcal{U}(0.6, 1.0)/\theta_p^a \sim \mathcal{U}(-0.3, -0.1)$, and Path 8 from $\theta_p^a \sim \mathcal{U}(-0.3, 0.9)/\theta_p^a \sim \mathcal{U}(0.6, 1.0)$ with $g_p \sim \mathcal{U}(0.001, 0.01)$.

In Fig. 5(c), we observe that the performance of our generative pipeline remains consistent across a varying number of paths P. This is because our model is independent of P, and leverages the formulation of the loss function in (12) to balance the reconstruction accuracy and the number of non-zero output values, which dictates the number of identified paths. The last term in (12), enforces output sparsity. The hyperparameter α_S in (12) can thereby be tuned by observing the reconstruction loss. Thus, our method can adapt to a range of values for the number of paths by finding a suitable balance between the NMSE and the sparsity loss such that the number of non-zero values are proportional to the number of paths.

5 Conclusion

In this paper, we developed a generative pipeline that leverages a PPGC model for parametrized channel generation. We tackle the extreme non-linearity in the model by developing a dictionary-based relaxation and learning a sparse gain matrix whose non-zero values denote the associated path parameters. We empirically show that our method captures path-specific parameter distributions for a given channel dataset and outperforms prior art in terms of 2-Wasserstein distance and MMD. Our work can be extended to 3-dimensional scenarios with angles of elevation and additional parameters such as delay.

Acknowledgements

This work was completed during Satyavrat Wagle's internship at InterDigital Communications.

Prof. Christopher G. Brinton and David J. Love helped to draft this article under support from the Office of Naval Research (ONR) grant N00014-21-1-2472, and the National Science Foundation (NSF) grants CNS-2212565 and ITE-2326898. Prof. Christopher G. Brinton was also supported in part by the Defense Advanced Research Projects Agency (DARPA) grant D22AP00168.

References

[Alkhateeb *et al.*, 2014] Ahmed Alkhateeb, Omar El Ayach, Geert Leus, and Robert W. Heath. Channel estimation and hybrid precoding for millimeter wave cellular systems. *IEEE Journal of Selected Topics in Signal Processing*, 8(5):831–846, 2014.

[Alkhateeb, 2019] A. Alkhateeb. DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications. In *Proc. of Information Theory and Applications Workshop (ITA)*, pages 1–8, San Diego, CA, Feb 2019.

[Arvinte and Tamir, 2022] Marius Arvinte and Jonathan Tamir. Score-based generative models for wireless channel modeling and estimation. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[Drakshayini and Kounte, 2022] MN Drakshayini and Manjunath R Kounte. A review of wireless channel estimation techniques: challenges and solutions. *Inter. Journal of Wireless and Mobile Computing*, 23(2):193–203, 2022.

[Dumoulin and Visin, 2016] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[Dziugaite *et al.*, 2015] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conf. on Uncertainty in Artificial Intelligence*, UAI'15, page 258–267, Arlington, Virginia, USA, 2015. AUAI Press.

[Forenza *et al.*, 2007] Antonio Forenza, David J. Love, and Robert W. Heath. Simplified spatial correlation models for clustered mimo channels with different array configurations. *IEEE Trans. on Vehicular Tech.*, 56(4):1924–1934, 2007.

[Hu *et al.*, 2022] Yaqi Hu, Mingsheng Yin, William Xia, Sundeep Rangan, and Marco Mezzavilla. Multi-frequency channel modeling for millimeter wave and thz wireless communication via generative adversarial networks. In *2022 56th Asilomar Conf. on Signals, Systems, and Computers*, pages 670–676, 2022.

[Huang *et al.*, 2019] Hao Huang, Yang Peng, Jie Yang, Wenchao Xia, and Guan Gui. Fast beamforming design via deep learning. *IEEE Trans. on Vehicular Tech.*, 69(1):1065–1069, 2019.

[Ju and Rappaport, 2023] Shihao Ju and Theodore S. Rappaport. 142 ghz multipath propagation measurements and path loss channel modeling in factory buildings. In *ICC 2023 - IEEE Inter. Conf. on Commun.*, pages 5048–5053, 2023.

[Ju *et al.*, 2022] Shihao Ju, Yunchou Xing, Ojas Kanhere, and Theodore S. Rappaport. Sub-terahertz channel measurements and characterization in a factory building. In *ICC 2022 - IEEE Inter. Conf. on Commun.*, pages 2882–2887, 2022.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2014.

[Kong *et al.*, 2017] Linghe Kong, Muhammad Khurram Khan, Fan Wu, Guihai Chen, and Peng Zeng. Millimeter-wave wireless commun. for iot-cloud supported autonomous vehicles: Overview, design, and challenges. *IEEE Commun. Mag.*, 55(1):62–68, 2017.

[Kullback and Leibler, 1951] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[Kumar *et al.*, 2024] Satyam Kumar, Akshay Malhotra, and Shahab Hamidi-Rad. Channel parameter estimation in wireless communication: A deep learning perspective. In *2024 IEEE Inter. Conf. on Mach. Learn. for Communication and Networking (ICMLCN)*, pages 511–516, 2024.

[Liang *et al.*, 2020] Peizhe Liang, Jiancun Fan, Wenhan Shen, Zhijin Qin, and Geoffrey Ye Li. Deep learning and compressive sensing-based csi feedback in fdd massive mimo systems. *IEEE Trans. on Vehicular Tech.*, 69(8):9217–9222, 2020.

[Mao *et al.*, 2018] Qian Mao, Fei Hu, and Qi Hao. Deep learning for intelligent wireless networks: A comprehensive survey. *IEEE Commun. Surveys & Tutorials*, 20(4):2595–2621, 2018.

[Mashhadi *et al.*, 2020] Mahdi Boloursaz Mashhadi, Qianqian Yang, and Deniz Gündüz. Distributed deep convolutional compression for massive mimo csi feedback. *IEEE Trans. on Wireless Commun.*, 20(4):2621–2633, 2020.

[Meijerink and Molisch, 2014] Arjan Meijerink and Andreas F. Molisch. On the physical interpretation of the saleh–valenzuela model and the definition of its power delay profiles. *IEEE Trans. on Antennas and Propagation*, 62(9):4780–4793, 2014.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th Inter. Conf. on Inter. Conf. on Mach. Learn.*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[Nguyen *et al.*, 2021] Ly V Nguyen, Duy HN Nguyen, and A Lee Swindlehurst. Dnn-based detectors for massive mimo systems with low-resolution adcs. In *ICC 2021-IEEE Inter. Conf. on Commun.*, pages 1–6. IEEE, 2021.

[Orekondy *et al.*, 2022] Tribhuvanesh Orekondy, Arash Behboodi, and Joseph B. Soriaga. Mimo-gan: Generative mimo channel modeling. In *ICC 2022 - IEEE Inter. Conf. on Commun.*, pages 5322–5328, 2022.

[Panaretos and Zemel, 2019] Victor M Panaretos and Yoav Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6(1):405–431, 2019.

[Qamar *et al.*, 2019] Faizan Qamar, MHD Nour Hindia, Kaharudin Dimyati, Kamarul Ariffin Noordin, Mohammed Bahjat Majed, Tharek Abd Rahman, and Iraj Sadegh Amiri. Investigation of future 5g-iot millimeter-wave network performance at 38 ghz for urban microcell outdoor environment. *Electronics*, 8(5):495, 2019.

[Remcom, 2025] Remcom. Wireless InSite. 2025. http://www.remcom.com/wireless-insite.

[Sant and Rao, 2024] Aditya Sant and Bhaskar D Rao. Insights into maximum likelihood detection for one-bit massive mimo commun. *IEEE Trans. on Wireless Commun.*, 2024.

[Sant *et al.*, 2022] Aditya Sant, Afshin Abdi, and Joseph Soriaga. Deep sequential beamformer learning for multipath channels in mmwave communication systems. In *ICASSP 2022-2022 IEEE Inter. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5198–5202. IEEE, 2022.

[Sengupta *et al.*, 2023] Ushnish Sengupta, Chinkuo Jao, Alberto Bernacchia, Sattar Vakili, and Da-shan Shiu. Generative diffusion models for radio wireless channel modelling and sampling. In *GLOBECOM 2023 - 2023 IEEE Global Commun. Conf.*, pages 4779–4784, 2023.

[Soltani *et al.*, 2019] Mehran Soltani, Vahid Pourahmadi, Ali Mirzaei, and Hamid Sheikhzadeh. Deep learning-based channel estimation. *IEEE Commun. Letters*, 23(4):652–655, 2019.

[Wagle *et al.*, 2025] Satyavrat Wagle, Akshay Malhotra, Shahab Hamidi-Rad, Aditya Sant, David J. Love, and Christopher G. Brinton. Physics-based generative models for geometrically consistent and interpretable wireless channel synthesis, 2025.

[Wen *et al.*, 2018] Chao-Kai Wen, Wan-Ting Shih, and Shi Jin. Deep learning for massive mimo csi feedback. *IEEE Wireless Commun. Letters*, 7(5):748–751, 2018.

[Xia *et al.*, 2022] William Xia, Sundeep Rangan, Marco Mezzavilla, Angel Lozano, Giovanni Geraci, Vasilii Semkin, and Giuseppe Loianno. Generative neural network channel modeling for millimeter-wave uav communication. *IEEE Trans. on Wireless Commun.*, 21(11):9417–9431, 2022.

[Xiao *et al.*, 2022] Han Xiao, Wenqiang Tian, Wendong Liu, and Jia Shen. Channelgan: Deep learning-based channel modeling and generating. *IEEE Wireless Commun. Letters*, 11(3):650–654, 2022.

[Yang *et al.*, 2019] Yang Yang, Yang Li, Wuxiong Zhang, Fei Qin, Pengcheng Zhu, and Cheng-Xiang Wang. Generative-adversarial-network-based wireless channel modeling: Challenges and opportunities. *IEEE Commun. Mag.*, 57(3):22–27, 2019.