PHYSICS-INFORMED GENERATIVE APPROACHES FOR WIRELESS CHANNEL MODELING

Satyavrat Wagle
Elmore School of Electrical and Computer Engineering, Purdue University
wagles@purdue.edu

Akshay Malhotra, Shahab Hamidi-Rad, Aditya Sant
InterDigital Communications
{firstname.lastname}@interdigital.com

David J. Love, Christopher G. Brinton
Elmore School of Electrical and Computer Engineering, Purdue University
{cgb,djlove}@purdue.edu

ABSTRACT

In recent years, machine learning (ML) methods have become increasingly popular in wireless communication systems for several applications. A critical bottleneck when 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 ML 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 of interest. In this work, we aim to address both these issues by integrating a parametric, physics-based geometric channel (PBGC) modeling framework with generative methods. To address limitations with gradient flow through the PBGC model, a linearized reformulation is presented, which ensures smooth gradient flow during generative model training, while also capturing insights about the underlying physical environment. We evaluate our model against prior baselines by comparing the generated samples in terms of the 2-Wasserstein distance and through the utility of generated data 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 systems, ML research primarily aims to estimate and mitigate distortions in electromagnetic signals during over-the-air (OTA) transmission Liang *et al.* (2020); Soltani *et al.* (2019); Mao *et al.* (2018); Sant *et al.* (2022) and address noise and non-linearities at transmit or receive antennas Drakshayini and Kounte (2022); Sant and Rao (2024). However, for practical deployments, the ML pipeline requires a substantial amount of OTA wireless channel data which is a complex, costly, and time-intensive Ju and Rappaport (2023); Ju *et al.* (2022); Kumar *et al.* (2024).

Generative models have been proposed to mitigate this problem by artificially synthesizing large wireless datasets using considerably fewer OTA data samples Xiao *et al.* (2022). However, unlike common modalities of data (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*. 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

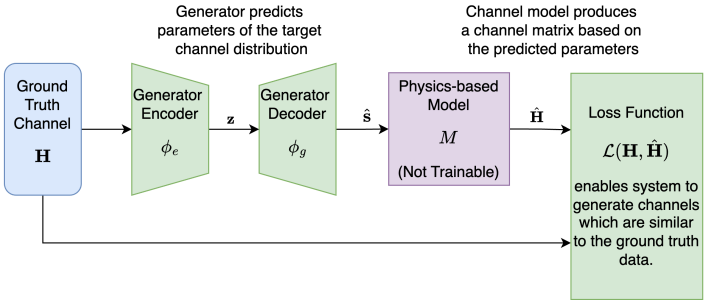


Figure 1: In the straightforward implementation, the generator directly predicts the parameters $\hat{\mathbf{s}}$, which are then used by the PBGC model M to produce the predicted channel.

the receiver Forenza *et al.* (2007); Meijerink and Molisch (2014). 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 models for synthesizing millimeter wave (mmWave) channels, crucial for next-generation wireless communication and IoT Kong *et al.* (2017). The proposed method overcomes the limitations of existing approaches by incorporating a physics-based generative channel (PBGC) model into the generative pipeline. As the PBGC model parametrizes the channel generation, our generative process will learn the joint distribution of the underlying parameters responsible for channel generation. We also propose a linearized reformulation of the PBGC model to address training challenges due to its non-convex loss landscape.

2 System Model and Approach

We consider a wireless communication system with N_t transmit and N_r receive antennas. The associated PBGC model defined by $M : \mathbb{R}^{3P} \to \mathbb{R}^{2\times N_t \times N_r}$ 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. The PBGC model is given by

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

Where, $\mathbf{s} = [g_p, \theta_a^p, \theta_d^p]_{p=1}^P$, and g_p represents the propagation gain associated with the p-th path, $\mathbf{a}_t(\cdot), \mathbf{a}_r(\cdot)$ represent the array response vectors on the transmit and receive antennas, θ_d^p and θ_a^p represent the corresponding angle of departure and angle of arrival. More details on the PBGC model can be found in Appendix B.

2.1 Generative Model to Predict Channel Statistics

For the generative model we use the variational autoencoder (VAE) architecture Kingma and Welling (2014), as seen in Fig. 1. We use the generative model to produce the parameter vector $\hat{\mathbf{s}}$ using a latent variable $\mathbf{z}$, which is then passed to the PBGC model M to produce a valid channel matrix. More details on the training can be found in Appendix C.

2.1.1 Limitations of generative model training using PBGC

The direct use of the PBGC model with the VAE leads to poor training performance due to the presence of sinusoidal functions in the channel model (Appendix B). This property is illustrated in Fig. 2. 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 Hinton (2010); Dumoulin and Visin (2016). As a result, depending on the location of the optimizer in

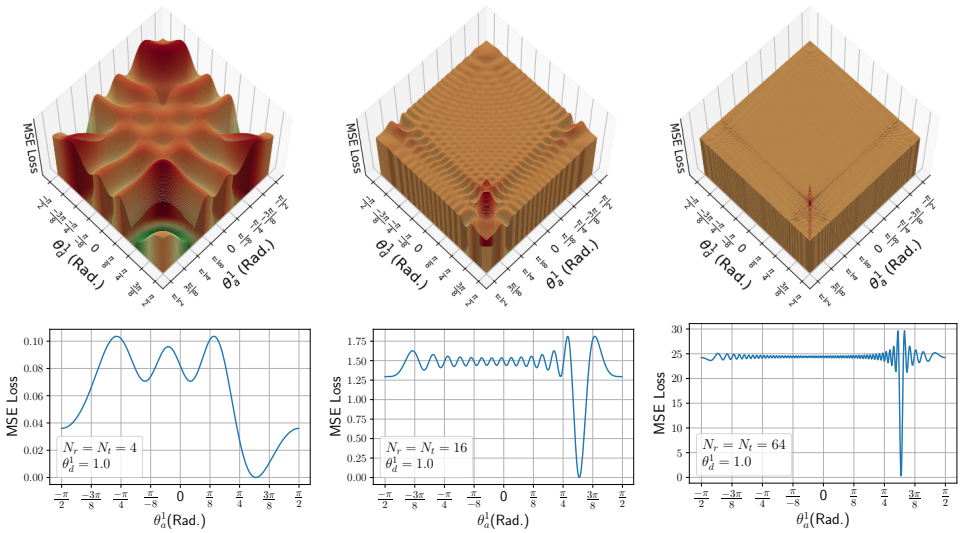


Figure 2: The loss surface as a function of (θ_a^1, θ_d^1) in reference to a channel matrix with $\theta_d^1 = \theta_a^1 = 1.0$ using a PBGC model M with $P = 1$ and $N_r = N_t = 4, 16, 64$ antennas respectively. The PBGC model M is extremely non-convex as a function of the parameters θ_a, θ_d because of periodicity arising from the formulation of the array response vectors. As the number of antennas N_r, N_t increase, gradient flowing through the model will be negligible when at locations further away from the minima, where the loss surface is effectively flat.

the parameter space, the gradient may not flow during backpropagation, resulting in the optimizer converging at non-optimal points. Further, as the number of antennas, N_t, N_r increase, the number of local minimas grows, and the optimality gap, the difference between the loss values at the local minima and the global minima, widens, significantly impacting the overall loss (Appendix C.1).

2.2 Linearized Reformulation of the Physics Model

To overcome the challenges posed by the PBGC model, while maintaining the key underlying model features for channel generation, the channel synthesis is presented via a weighted sum over a parameter dictionary. In order to compute the dictionary elements, the range of angles θ_a^p, θ_d^p, given by $[\theta_{\min}, \theta_{\max}]$, is 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, where each element of the dictionary, $\mathbf{D}_{i,j}\ \forall\, i, j \in \{1, R\}$, is given by:

$$\mathbf{D}_{i,j} = \mathbf{a}_r(\theta_i)\mathbf{a}_t(\theta_j). \tag{2}$$

The relaxed PBGC model using the linearized reformulation can now be expressed as,

$$\mathbf{H} = \sum_{i=1}^{R}\sum_{j=1}^{R} \mathbf{W}_{i,j}\mathbf{D}_{i,j}, \tag{3}$$

where, the channel generation is parametrized by the gain matrix $\mathbf{W} \in \mathbb{R}^{R^2}$, instead of the parameters $\mathbf{s}$, and the 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 PBGC model M. Further, it must be noted that the accuracy of the relaxed model is strongly tied to the number of intervals, R. For a suitably high value of R, any channel $\mathbf{H}$ can be approximated by (3).

2.3 Generative Model to Predict the Gain Matrix

The modified generative model using the linearized reformulation is shown in Fig. 3. 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 $\mathbf{W} = g_{\phi_d}(\mathbf{z})$. The gain matrix is then used to generate the predicted channel using equation 3.

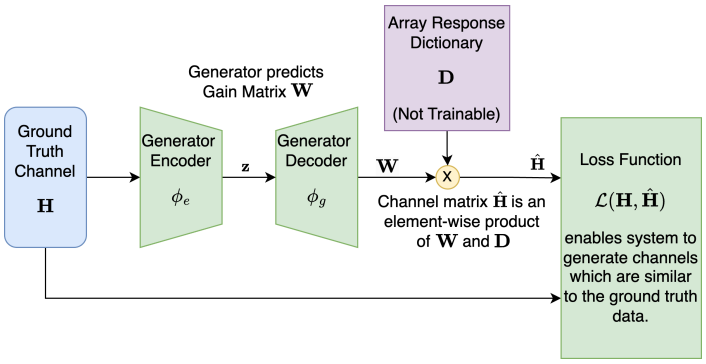


Figure 3: We relax the PBGC model by defining a discretized array response dictionary **D** and using the generator to output gain the gain matrix **W**. The elementwise multiplication of **W** and **D** mimics the PBGC model process. This relaxation allows the flow of gradient through the generator,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

	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 1: 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).

3 Experimental Results

In this section, we analyze the performance of our method on wireless datasets generated corresponding to real-life scenarios and compare our method against prior art baselines. For our PBGC model equation 1, we consider transmit and receive antennas $N_t = N_r = 16$ and assume the number of paths $P = 5$. We consider five datasets, sythesized using the DeepMIMO framework Alkhateeb (2019), modeling street intersections in different cities (see Appendix D.1). We use a VAE as the generative model, $(g_{\phi e}, g_{\phi d})$, which is trained using the Adam optimizer with a learning rate of $1e^{-3}$. Model architecture details are described in Appendix C 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. For our model, we use resolution $R = 64$ and latent dimension $z = 64$.

In Table 1, 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 Zemel (2019) 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. A comprehensive evaluation of the generative modeling schemes on the downstream task of channel compression (CSI compression) can be found in Appendix D.

4 Conclusion

In this paper, we developed a generative pipeline that leverages a PBGC model for parametrized channel generation. We tackle the extreme non-convexity in the PBGC model by developing a dictionary-based relaxation of the PBGC model and learning a sparse gain matrix whose non-zero values denote the parameters of the associated paths. We empirically show that our method effectively captures path-specific parameter distributions for a given dataset of channel matrices and outperforms prior arts 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 path delay.

5 ACKNOWLEDGEMENTS

This project was undertaken as part of an internship at Interdigital, Inc.

Prof. Christopher G. Brinton's contributions to this project were supported in part by the Defense Advanced Research Projects Agency (DARPA) under grant D22AP00168, the Office of Naval Research (ONR) under grant N00014-21-1-2472, and the National Science Foundation (NSF) under grant CNS-2212565

REFERENCES

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.

Ahmed 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.

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.

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

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

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

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

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 Conference on Signals, Systems, and Computers*, pages 670–676, 2022.

Shihao Ju and Theodore S. Rappaport. 142 ghz multipath propagation measurements and path loss channel modeling in factory buildings. In *ICC 2023 - IEEE International Conference on Communications*, pages 5048–5053, 2023.

Shihao Ju, Yunchou Xing, Ojas Kanhere, and Theodore S. Rappaport. Sub-terahertz channel measurements and characterization in a factory building. In *ICC 2022 - IEEE International Conference on Communications*, pages 2882–2887, 2022.

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

Linghe Kong, Muhammad Khurram Khan, Fan Wu, Guihai Chen, and Peng Zeng. Millimeter-wave wireless communications for iot-cloud supported autonomous vehicles: Overview, design, and challenges. *IEEE Communications Magazine*, 55(1):62–68, 2017.

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

Satyam Kumar, Akshay Malhotra, and Shahab Hamidi-Rad. Channel parameter estimation in wireless communication: A deep learning perspective. In *2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, pages 511–516, 2024.

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 Transactions on Vehicular Technology*, 69(8):9217–9222, 2020.

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

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

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

Tribhuvanesh Orekondy, Arash Behboodi, and Joseph B. Soriaga. Mimo-gan: Generative mimo channel modeling. In *ICC 2022 - IEEE International Conference on Communications*, pages 5322–5328, 2022.

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

Remcom. Wireless InSite. `http://www.remcom.com/wireless-insite`.

Aditya Sant and Bhaskar D Rao. Insights into maximum likelihood detection for one-bit massive mimo communications. *IEEE Transactions on Wireless Communications*, 2024.

Aditya Sant, Afshin Abdi, and Joseph Soriaga. Deep sequential beamformer learning for multi-path channels in mmwave communication systems. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5198–5202. IEEE, 2022.

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 Communications Conference*, pages 4779–4784, 2023.

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

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

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 Transactions on Wireless Communications*, 21(11):9417–9431, 2022.

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

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 Communications Magazine*, 57(3):22–27, 2019.

A 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). Authors of 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. 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), the authors utilize diffusion models as the generative backbone to circumvent the issue of mode collapse in GANs. Works such as Arvinte and

Tamir (2022) utilize a score-based generative model for joint channel modeling and estimation. All of the above works utilize a generative model to directly produce channel matrices at the output have no guarantees on the validity of the generated channel data, with limited interpretability.

An alternative research direction involves using labeled datasets to predict parameters associated with the wireless channel. In Xia *et al.* (2022) and Hu *et al.* (2022), the authors use a conditional variational autoencoder (VAE) and a conditional GAN framework, respectively, to generate the channel parameters given the location of UAVs. The above works require datasets labeled with metadata relating to the environment and locations of the transmitter or receiver to produce parameters that can be used in downstream channel models and do 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.

B Details of the Physics Based Geometric Channel Model (PBGC)

As mentioned, we consider a communication setup with N_t transmit and N_r receive antennas. The PBGC model considers the channel matrix $\mathbf{H}$ to be a superposition of the individual propagation matrices associated with each of the P paths. The overall model is thus expressed as:

$$\mathbf{H} = M(\mathbf{s}) = \sum_{p=1}^{P} g_p \mathbf{a}_r(\theta_a^p)\mathbf{a}_t(\theta_d^p)^H. \tag{4}$$

Where, $\mathbf{s} = [g_p, \theta_a^p, \theta_d^p]_{p=1}^P$, and g_p represents the propagation gain associated with the p-th path, $\mathbf{a}_t(\cdot), \mathbf{a}_r(\cdot)$ represent the array response vectors on the transmit and receive antennas, 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. Also, in mmWave channels, the total number of paths P is typically small in over-the-air transmission.

To simplify the discussion, we consider Uniform Linear Arrays (ULA) Forenza *et al.* (2007) and limit the discussion to the azimuth plane. Thus, the array response vectors can be defined as:

$$\mathbf{a}_t(\theta_d^p) = \frac{1}{\sqrt{N_t}}[1, e^{ju\sin(\theta_d^p)}, \ldots, e^{j(N_t-1)u\sin(\theta_d^p)}]^T, \tag{5}$$

$$\mathbf{a}_r(\theta_a^p) = \frac{1}{\sqrt{N_r}}[1, e^{ju\sin(\theta_a^p)}, \ldots, e^{j(N_r-1)u\sin(\theta_a^p)}]^T, \tag{6}$$

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

C Generative Model Training with PBGC

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 $\mathbf{z} \sim f_{\phi_e}(\mathbf{H})$. 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 $\hat{\mathbf{s}} = g_{\phi_d}(\mathbf{z})$. The predicted parameter vector is then passed to the model M to produce an output channel $\hat{\mathbf{H}}$ as follows $\hat{\mathbf{H}} = M(\hat{\mathbf{s}})$. The system loss is a generalization of the evidence based lower bound (ELBO)Kingma and Welling (2014), given by

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

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 Kulback-Leibler (KL) divergence Kullback and Leibler (1951) 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 $\mathcal{N}(0, \mathbf{I})$.

C.1 Convergence Issues with the PBGC Model

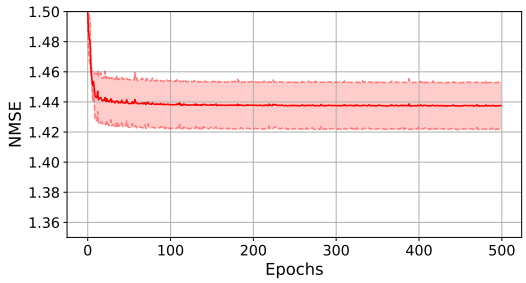


Figure 4: In the straightforward integration of the PBGC model with the VAE, the generator directly predicts the parameters $\hat{\mathbf{s}}$, which are then used by the PBGC model M to produce the predicted channel. In such implementations, the generator cannot converge to suitable optima due to the non-convexity of the PBGC model. Further, as the number of antennas, N_t, N_r increases, the optimality gap, the difference between the loss values at the local minima and the global minima, widens, significantly impacting the overall loss. In this figure, we show the average training performance of the pipeline over several independent training instances. As can be seen, the normalized mean squared error (NMSE) barely reduces and continues to remain high across the entire training horizon.

C.2 Model Architecture

We consider a variational autoencoder (VAE) architecture as the generative model for our experiments. The architecture of our model is given in Fig. 5.

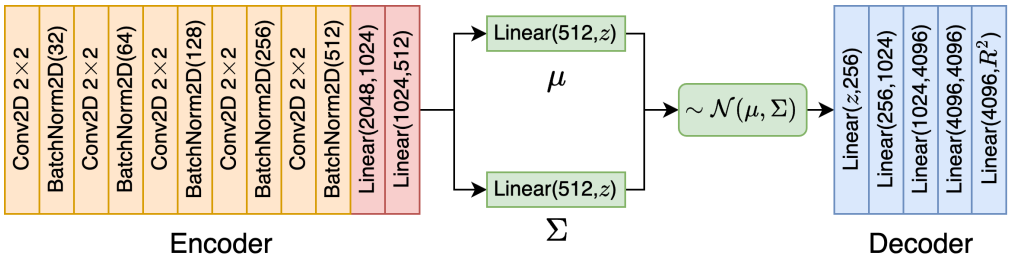


Figure 5: The architecture of the VAE model used in our experiments. We use the Leaky ReLU activation after each BatchNorm and Linear layer.

D Experimental Results

D.1 Dataset Information

Five datasets are generated using the used to generate channel datasets using 3D ray tracing. Datasets corresponding to the following scenarios are used; (i) Two base stations in an outdoor intersection of two streets with blocking and reflecting surfaces, given by (BS10) and (BS11); (ii) an indoor conference room, given by (Indoor); (iii) a section of downtown Boston, Massachusetts, USA, generated using the 5G model developed by RemCom Remcom, given by (Boston) and (iv) a section of the Arizona State University campus in Tempe, Arizona, USA, given by (ASU).

We evaluate the accuracy of generated channels as well as effectiveness of the generative pipeline for downstream channel compression tasks.

D.2 Prediction of Parameters

We evaluate the accuracy of our system in capturing the underlying distribution of parameters associated with a given dataset. A user-defined dataset $\mathbf{D}$ of channel matrices is created by sampling

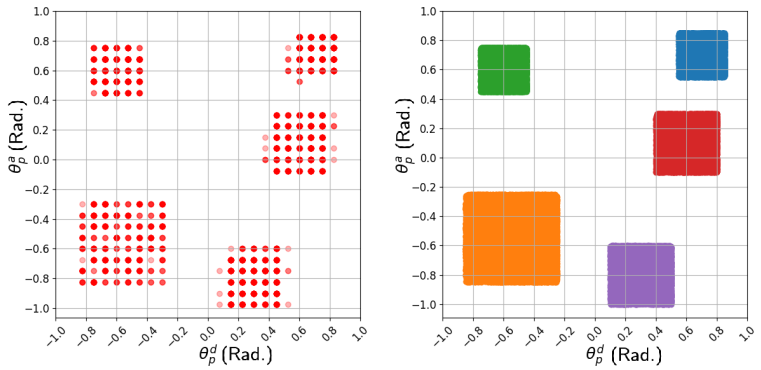
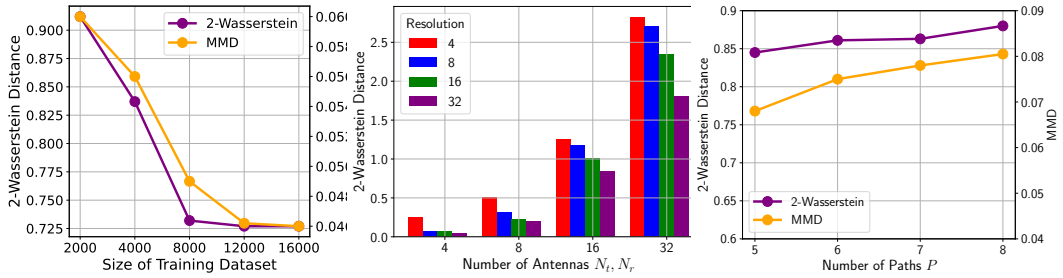


Figure 6: The distributions of angles of arrival and departure (θ_a^p, θ_d^p) captured by our method (Left) match the underlying distributions of the training dataset (Right). Each color corresponds to a distinct path.



(a) (b) (c)

Figure 7: (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 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 $\mathbf{W}$.

parameters $[g_p, \theta_p^a, \theta_p^d]_{p=1}^P$ from a user-defined distribution. The proposed model is trained on this dataset, and the distribution of the generated parameters from this model is illustrated. The results Fig. 6, compare the ground truth distributions with the output of the generative model. We observe that our method can accurately capture the distributions of the angles of arrival and departure for each path. 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.

D.3 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. 7(a), we observe that our method is able to provide a similar level 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.

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 2: 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.

D.4 Effect of Increasing Number of Paths

In this experiment, we observe the effect of increasing the number of constituent paths P in a channel matrix. We consider the channel dataset given in Table 1 and add additional paths where needed 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 is sampled from $\theta_p^a \sim \mathcal{U}(0.6, 1.0)/\theta_p^a \sim \mathcal{U}(-0.3, -0.1)$, and Path 8 is sampled from $\theta_p^a \sim \mathcal{U}(-0.3, 0.9)/\theta_p^a \sim \mathcal{U}(0.6, 1.0)$. $g_p \sim \mathcal{U}(0.001, 0.01)$ for all paths.

In Fig. 7(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 to balance the reconstruction accuracy and the number of non-zero output values, which dictates the number of identified paths. This is encapsulated by the last term, which enforces output sparsity. The hyperparameter α_S in the loss function 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 L1 regularization loss such that the number of non-zero values are proportional to the number of paths.

D.5 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 distinct channel distributions in the context of a downstream channel compression task.

We consider two channel datasets R_{10}, R_{11} from the DeepMIMO 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 2.

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 2, 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.