Learning Beam Codebooks with Neural Networks: Towards Environment-Aware mmWave MIMO

Yu Zhang, Muhammad Alrabeiah, and Ahmed Alkhateeb
School of Electrical, Computer, and Energy Engineering
Arizona State University
Email: y.zhang, malrabei, alkhateeb@asu.edu

***Abstract*—Scaling the number of antennas up is a key characteristic of current and future wireless communication systems. The hardware cost and power consumption, however, motivate large-scale MIMO systems, especially at millimeter wave (mmWave) bands, to rely on analog-only or hybrid analog/digital transceiver architectures. With these architectures, mmWave base stations normally use pre-defined beamforming codebooks for both initial access and data transmissions. Current beam codebooks, however, generally adopt single-lobe narrow beams and scan the entire angular space. This leads to high beam training overhead and loss in the achievable beamforming gains. In this paper, we propose a new machine learning framework for learning beamforming codebooks in hardware-constrained large-scale MIMO systems. More specifically, we develop a neural network architecture that accounts for the hardware constraints and learns beam codebooks that adapt to the surrounding environment and the user locations. Simulation results highlight the capability of the proposed solution in learning multi-lobe beams and reducing the codebook size, which leads to noticeable gains compared to classical codebook design approaches.**

***Index Terms*—Machine learning, beamforming codebooks, environment awareness, neural networks, large-scale MIMO.**

I. Introduction

Millimeter wave (mmWave) and Terahertz (THz) communication systems are essential components of 5G and beyond. To guarantee sufficient received signal power, these systems deploy large antenna arrays and use narrow beams. Due to the high cost and power consumption of high frequency RF chains, however, mmWave and THz communication systems can not use fully-digital architectures that assign an RF chain per antenna, and normally resort to analog-only or hybrid analog/digital architectures [1], [4]. Further, because of the hardware constraints on these large-scale MIMO systems, they typically adopt pre-defined single-lobe beamforming codebooks (such as DFT codebooks [2]) that scan all possible directions for both initial access and data transmission. These codebooks have two key limitations: (i) They incur high beam training overhead by scanning all possible directions even though many of these directions may never be used, and (ii) they normally have single-lobe beams which may not be optimal, especially in non-line-of-sight (NLOS) scenarios.

Contribution: In this paper, we consider hardware-constrained large-scale MIMO systems and propose an artificial neural network based framework for learning environment aware beamforming codebooks. More specifically, we design a machine learning model that can learn how to adapt the patterns of the codebook beams based on the surrounding environment and user distribution. This is done by developing a novel complex-valued neural network architecture in which the weights directly model the beamforming/combining weights. Further, the developed neural network architecture accounts for the hardware constraints (such as phase-only, constant-modulus, and quantized-angle constraints) that are typically applied on the analog-only mmWave/THz transceiver architectures. We evaluate the performance of the proposed solution in both outdoor LOS and indoor NLOS communication scenarios. The simulation results highlight the capability of the proposed solution in learning multi-lobe beams that adapt to the environment geometry and to where the users are. Further, for the adopted NLOS simulation setup, the results show that the learned codebook can exceed the 64-beam DFT codebook with only 16 beams, which can lead to noticeable savings in the beam training overhead.

Prior work: Designing beamforming codebooks has been an important research topic for long time at both academia and industry (see [3] for a good overview). With large-scale MIMO systems, however, the hardware limitations (especially at mmWave/THz) and the use of analog-only or hybrid transceiver architectures imposed new constraints on the codebook design problems. This motivated the development of new beamforming codebooks [1], [2], [4], [5]. These codebooks, however, are generally designed to have single-lobe narrow beams that cover all the angular directions and are not adaptive to the deployment, surrounding environment, and user distributions. This motivated the development of environment and hardware aware codebook learning strategies, which is the focus of this paper.

II. System and Channel Models

In this paper, we consider a system model where a mmWave base station (BS), equipped with M antennas, is communicating with a single-antenna user. In the uplink transmission, if the user transmits a symbol $x \in \mathbb{C}$ to the BS, with the power constraint $\mathbb{E}\left[|x|^2\right] = P$, the received signal at the BS after combining can be expressed as

$$y = \mathbf{w}^H \mathbf{h} x + \mathbf{w}^H \mathbf{n}, \tag{1}$$

where $\mathbf{h} \in \mathbb{C}^{M \times 1}$ is the uplink channel vector between the mobile user and the BS, $\mathbf{w}$ is the receive combining vector at the BS and $\mathbf{n} \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma_n^2 \mathbf{I}\right)$ is the receive noise vector at

978-1-7281-5478-7/20/$31.00 ©2020 IEEE

the BS. To reduce the cost and power consumption, the BS is assumed to employ analog-only transceiver architecture [6], where the beamforming/combining processing is implemented using analog phase shifters. To account for that, each element w_m in the combining vector, $m = 1, 2, ..., M$, is assumed to have the structure $w_m = \frac{1}{\sqrt{M}} e^{j\theta_m}$.

We adopt a general geometric channel model for $\mathbf{h}$ [7]. Assume that the signal propagation between the mobile user and the BS consists of L paths. Each path ℓ has a complex gain α_ℓ and an angle of arrival ϕ_ℓ. Thus, the channel can be written as

$$\mathbf{h} = \sum_{\ell=1}^{L} \alpha_\ell \mathbf{a}(\phi_\ell), \tag{2}$$

where $\mathbf{a}(\phi_\ell)$ is the array response vector of the BS. The definition of $\mathbf{a}(\phi_\ell)$ depends on the array geometry.

III. Problem Definition

In this paper, we consider hardware-constrained large-scale MIMO systems and investigate the design of beamforming codebooks that are adaptive to the deployment scenario, environment geometry, user distributions, etc.—which we refer to as environment-aware beamforming codebooks.

Given the system and channel models in Section II, and due to the hardware constraints on the BS transceiver, the BSs normally adopt pre-defined beamforming codebooks. If $\boldsymbol{\mathcal{W}} = \{\mathbf{w}_1, ..., \mathbf{w}_N\}$ denotes the BS beamforming codebook, then the maximum beamforming/combining gain of user u can be written as

$$\gamma_u = \max_{\mathbf{w}_n \in \boldsymbol{\mathcal{W}}} \left| \mathbf{w}_n^H \mathbf{h}_u \right|^2. \tag{3}$$

Our objective in this paper is then to design the beamforming codebook $\boldsymbol{\mathcal{W}}$ to maximize the maximum beamforming gain γ_u averaged over the set of users served by this BS in the surrounding environment. Let $\mathcal{H}$ represent the set of channel vectors for the candidate users, then our objective can be written as

$$\boldsymbol{\mathcal{W}}_{\text{opt}} = \arg\max_{\boldsymbol{\mathcal{W}}} \frac{1}{|\mathcal{H}|} \sum_{\mathbf{h}_u \in \mathcal{H}} \left[\max_{\mathbf{w}_n \in \boldsymbol{\mathcal{W}}} \left| \mathbf{w}_n^H \mathbf{h}_u \right|^2 \right], \tag{4}$$

$$\text{s.t. } |w_{mn}| = \frac{1}{\sqrt{M}}, \ \forall m \in [M], \ \forall n \in [N], \tag{5}$$

where $|\boldsymbol{\mathcal{W}}| = N$ is the codebook size, and $[N]$ is the shorthand for representing set $\{1, 2, \ldots, N\}$. The constraint (5) is imposed to uphold the phase-shifters constraint.

Constructing efficient beamforming codebooks has attracted significant interest from both academia and industry for the last two decades (see [3] for a good overview). With large-scale MIMO systems, however, the hardware limitations, such as the constant modulus captured by (5) and the quantized phase shifters, add hard and non-convex constraints to the beam codebook optimization problems and make them difficult to solve [4]. In this paper, we make an initial step into leveraging the learning and optimization capabilities of neural networks to learn beam codebooks that are adaptive to the environment geometry, user distributions, etc., and that respect the hardware constraints on the large-scale MIMO systems. We believe that this direction of using neural networks to develop environment and hardware aware beam codebooks could lead in the future extensions to interesting solutions, especially when considering more complex system models such as interference-limited and multi-user MIMO.

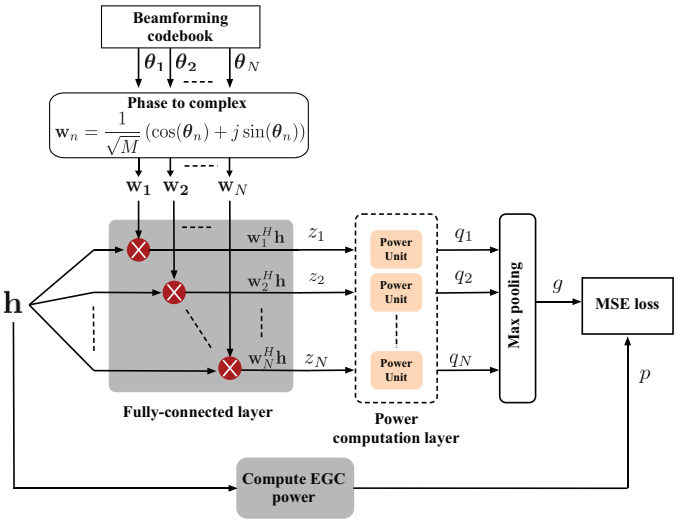


Fig. 1. This schematic shows the overall architecture of the neural network used to learn beamforming codebooks. It highlights the network architecture and the auxiliary components, equal-gain-combining and MSE-loss units, used during the training process. It also gives a slightly deeper dive into the inner-workings of the cornerstone of this architecture, the complex-valued fully-connected layer.

IV. Proposed Machine Learning Solution

In this section, we present our proposed beam codebook learning solution for hardware-constrained large-scale MIMO systems. First, we explain in Section IV-A the proposed neural network architecture and its relation to the optimization problem (4)-(5), before going into the details of how a codebook is learned in Section IV-B.

A. Model Architecture

This proposed neural network architecture consists of three main components, as depicted in Fig. 1. Those components are the complex-valued fully-connected layer, the power-computation layer and the max-pooling layer. A forward pass through these three layers is equivalent to evaluating the objective function of (4) over a single user's channel $\mathbf{h}$.

1) Complex-valued fully-connected layer: The first layer consists of N neurons that are capable of performing complex-valued multiplication and summation. Each neuron, as shown in Fig. 1, learns one beamforming vector and performs inner product with the input channel vector. Formally, this is described by the following matrix multiplication

$$\mathbf{z} = \mathbf{W}^H \mathbf{h}, \tag{6}$$

where $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_N] \in \mathbb{C}^{M \times N}$ is the beamforming codebook matrix, $\mathbf{h} \in \mathbb{C}^{M \times 1}$ is a user's channel vector, and $\mathbf{z} \in \mathbb{C}^{N \times 1}$ is the vector of the combined received signal.

(6) could be re-written in the following equivalent real-valued block matrix form

$$\begin{bmatrix} \mathbf{z}^r \\ \mathbf{z}^{im} \end{bmatrix} = \begin{bmatrix} \mathbf{W}^r & -\mathbf{W}^{im} \\ \mathbf{W}^{im} & \mathbf{W}^r \end{bmatrix}^T \begin{bmatrix} \mathbf{h}^r \\ \mathbf{h}^{im} \end{bmatrix}, \tag{7}$$

where $\mathbf{z}^r$, $\mathbf{z}^{im}$ are the real and imaginary parts of $\mathbf{z}$, $\mathbf{W}^r$, $\mathbf{W}^{im}$ are matrices containing the real and imaginary components of the elements of $\mathbf{W}$, and similarly, $\mathbf{h}^r$, $\mathbf{h}^{im}$ are the real and imaginary components of the channel vector $\mathbf{h}$.

Contrary to the norm in designing neural networks, the elements of the beamforming matrix $\mathbf{W}$ are not the weights of the fully-connected layer. Instead, they are derived from the actual neural network weights, which are the phases of the phased arrays. This is done through an embedded layer of phase-to-complex operations, as shown in Fig. 1. This layer transforms the phased arrays into unit-magnitude complex vectors by applying elements-wise $\cos$ and $\sin$ operations and scaling them by $1/\sqrt{M}$ as follows

$$\mathbf{W} = \frac{1}{\sqrt{M}} \left(\cos\left(\mathbf{\Theta}\right) + j \sin\left(\mathbf{\Theta}\right) \right), \tag{8}$$

where $\mathbf{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N]$ is an $M \times N$ matrix that includes the phases of the beamforming codebook, and $\boldsymbol{\theta}_n = [\theta_{1n}, \ldots, \theta_{Mn}]^T$, $\forall n \in \{1, \ldots, N\}$ is a single phase vector. The use of this embedded layer is the network's way of learning beamforming vectors that respect the phase shifter constraint.

2) Power-computation layer: The output of the complex-valued fully-connected layer is then fed into the power-computation layer. It performs element-wise absolute square operation and outputs a real-valued vector $\mathbf{q}$ given by

$$\mathbf{q} = [q_1, q_2, \ldots, q_N]^T = \left[|z_1|^2, |z_2|^2, \ldots, |z_N|^2 \right]^T, \tag{9}$$

which contains the received power of each beamforming vector in the codebook.

3) Max-pooling layer: The power of the best beamforming vector is finally found by the last layer, the max-pooling layer. It performs the following element-wise $\max$ operation over the elements of $\mathbf{q}$, that is

$$g = \max\left\{ q_1, q_2, \ldots, q_N \right\}, \tag{10}$$

where g is the power of the best beamforming vector. This value is used to assess the quality of the codebook by comparing it to a desired receive power value. In the following subsection, we describe the details of the desired value and how the quality is assessed.

B. Learning Codebooks

With the learning architecture in mind, it is time to delve into the details of how a codebook is learned. Our proposed solution follows a supervised learning approach. In such approach, a machine learning model is trained using pairs of inputs and the desired responses, which constitute the dataset.

1) Dataset Description: The inputs to the model are the users' channels as they are the communication quantity that drives the beamforming design process. For labels, there are many possible desired responses that could be used for training, and the choice between them should be made based on what the model needs to learn. In this paper, we adopt the equal gain combining (EGC) to construct the model label. This choice is based on the facts that EGC respects the phase shifters constraint and it is the beamforming strategy that achieves optimal beamforming gain when there are no restrictions on the codebook size or the quantized phase shifters. The EGC beamforming vector is constructed by using the phase component of every user's channel as $\mathbf{f}_{\mathrm{EGC}} = \frac{1}{\sqrt{M}} e^{\angle \mathbf{h}}$, where $\angle \mathbf{h}$ stands for the vector of phases of a complex vector $\mathbf{h}$. Using EGC beamforming vectors, the desired response for each user can be computed as follows

$$p = \left| \mathbf{f}_{\mathrm{EGC}}^H \mathbf{h} \right|^2 = \frac{1}{M} \left\| \mathbf{h} \right\|_1^2, \tag{11}$$

where $\| \cdot \|_1$ is the L_1 norm of a vector. Putting the users' channels and their EGC gains together provides the training dataset $\mathcal{S}_t$. This is formally given by

$$\mathcal{S}_t = \left\{ (\mathbf{h}_1, p_1), \ldots, (\mathbf{h}_U, p_U) \right\}, \tag{12}$$

where $\mathbf{h}_u \in \mathbb{C}^{M \times 1}$ and $p_u \in \mathbb{R}$ are the channel vector and EGC gain of the u-th user respectively, and $U = |\mathcal{H}|$ is the total number of collected data pairs.

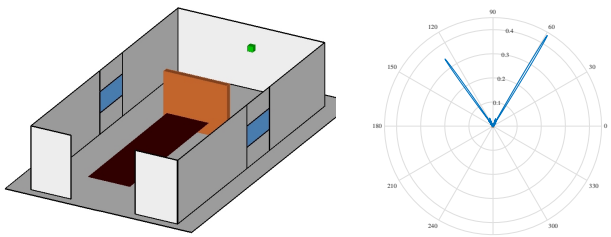
2) Model Training: Using the set $\mathcal{S}_t$, the model is trained by undergoing multiple forward-backward cycles. In each cycle, a *mini-batch* of complex channel vectors and their EGC responses is sampled from the training set. The channels are fed sequentially to the model and a forward pass is performed as described in Section IV-A. For each channel vector in the batch, the model combines it with the currently available N beamforming vectors and outputs the power of the best beamforming vector for that channel. The quality of the best combiner is assessed by measuring how close its beamforming gain to that of the equal gain combiner, obtained by (11). A Mean Squared Error (MSE) loss is used as a metric to assess the quality of the codebook over the current mini-batch. Formally, it is defined as

$$\mathcal{L} = \frac{1}{B} \sum_{b=1}^{B} (g_b - p_b)^2, \tag{13}$$

where g_b is the output of the max-pooling layer for the b-th data pair in the mini-batch, p_b is the desired value achieved by (11) and B is the mini-batch size.

The error signal (derivative of the loss (13) with respect to each phase vector $\boldsymbol{\theta}_n \in \mathbf{\Theta}$) is back propagated through the model to adjust the phases of the combining vectors [8] [9], making up what is usually referred to as the backward pass or *backpropagation*. This is formally expressed by the chain rule of differentiation

$$\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_n} \right)^T = \frac{\partial \mathcal{L}}{\partial g} \cdot \left(\frac{\partial g}{\partial \mathbf{q}} \right)^T \cdot \frac{\partial \mathbf{q}}{\partial \mathbf{z}} \cdot \frac{\partial \mathbf{z}}{\partial \boldsymbol{\theta}_n}. \tag{14}$$



(a) (b)

Fig. 2. The demonstration of how the learned codebook is environment-aware. (a) shows the NLOS scenario. It is chosen to be indoor for the high likelihood of having NLOS connection there. (b) is selected from the 64-beam codebook learned for the scenario in (a). It shows one of the beams with multi-lobes adapted for the different NLOS paths.

Mathematically, $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_n}$ does not exist for two reasons: 1) the factor $\frac{\partial g}{\partial \mathbf{q}}$ does not exist since the max-pooling function g is not differentiable, and 2) the factor $\frac{\partial \mathbf{q}}{\partial \mathbf{z}}$ does not satisfy the Cauchy-Riemann equations [10], meaning that $\mathbf{q}$ as a function of complex vector $\mathbf{z}$ is not complex differentiable (*holomorphic*). However, both issues could be resolved to enable backpropagation. The details of that are discussed in [11]. Computing the partial derivative of loss with respect to phase vector $\boldsymbol{\theta}_n$ allows the backward pass to modify the codebook $\boldsymbol{\Theta}$ and make it adaptive to the environment. The update equation generally depends on the solver used to carry out the training, e.g., Stochastic Gradient Descent (SGD) and ADAptive Moment estimation (ADAM) to name two, but in its simplest form, it could be given by

$$\boldsymbol{\theta}_{n_{\text{new}}} = \boldsymbol{\theta}_{n_{\text{old}}} - \eta \cdot \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_n}, \tag{15}$$

where η is the optimization step size, commonly known as the learning rate in machine learning, and $\boldsymbol{\theta}_{n_{\text{new}}}$, $\boldsymbol{\theta}_{n_{\text{old}}}$ are the new and current n-th phase vector of the codebook respectively.

Through the forward-backward cycles, the model learns to adapt its beamforming vectors according to the channels it experiences in the mini-batch. As mentioned in Section IV-A, a forward pass is equivalent to evaluating the objective function in (4). In a similar spirit, the backward pass through the architecture independently *optimizes* the beamforming vectors based on the **group of channels** that they receive best. It turns out that such group of channels that are best received by the same beamforming vector contribute only to the update of that beamforming vector and do not affect the others. This mainly comes as a result of the masking role that the max-pooling layer plays in the backward pass.

V. Experimental Setup and Model Training

In this section, we describe the adopted scenarios, datasets and the machine learning parameters used in our simulation.

A. Scenario and Dataset

The performance of the proposed solution is evaluated on two different communication scenarios. The first is an outdoor scenario which addresses the situation when all users experience LOS connection with the base station, named LOS scenario. The second, on the other hand, is an indoor scenario and addresses the case when none of the users has a LOS connection, which is referred to as NLOS scenario. Both scenarios are for an operating frequency of 28GHz, and both are part of the scenarios in the DeepMIMO dataset [12] . Using the data generation script of DeepMIMO, two sets of channel data are generated, one for each scenario. Table I shows the parameters of channel data generation.

TABLE I
The parameters of the DeepMIMO dataset

Name of scenario	O1_28	I2_28B
Active BS	3	1
Active users	700 to 1300	1 to 700
Number of antennas (x, y, z)	(1, 64, 1)	(64, 1, 1)
System BW	0.2 GHz	0.2 GHz
Antenna spacing	0.5	0.5
Number of OFDM sub-carriers	1	1
OFDM sampling factor	1	1
OFDM limit	1	1

B. Model Training and Testing

The model is trained and tested by using both datasets described in Section V-A. Prior to any training session, the channels of each dataset is pre-processed to improve the training experience [9], which is a very common practice in machine learning. The pre-processing of choice for these experiments is data sample normalization. As in [7], [13], the channel normalization using the maximum absolute value in the training dataset helps the network undergo a stable and efficient training. Formally, the normalization factor is found as follows

$$\Delta = \max_{\mathbf{h}_u \in S_t} |[\mathbf{h}_u]_m|^2 \tag{16}$$

where $[\mathbf{h}_u]_m$ is the m-th element in the channel vector of the $\mathbf{h}_u$. Using the normalized channels, the model is trained on portion of the samples of the dataset and tested on the rest. For brevity, the data split percentage between training and testing along with other training hyper-parameters can be found at [14].

VI. Simulation Results

In this section, we evaluate the performance of our proposed machine learning based beamforming codebook design solution on the scenarios described in Section V-A. The numerical results show that our proposed model can achieve performance that is comparable to EGC and adapt to different user distributions.

Fig. 3 shows the evaluation results of the proposed solution in a LOS scenario. With 64 beams, the learned codebook reaches more than 90% of the upper bound, which is achieved by the EGC where the number of beams in the codebook is equal to the number of users (and assuming unquantized phase shifters). Further, with 128 beams, the developed approach achieves more than 95% of the upper bound. Besides, Fig. 3 also exhibits the performance of the codebook with quantized

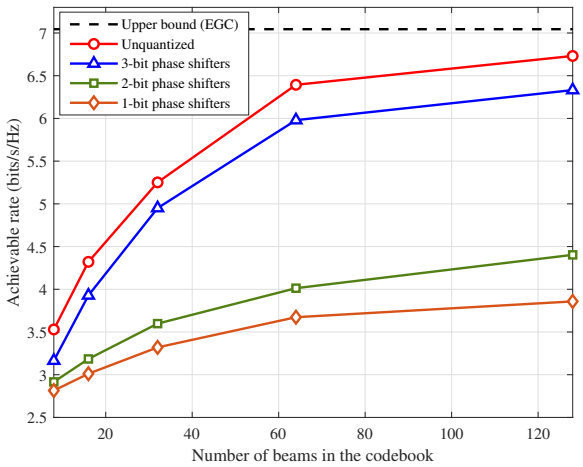


Fig. 3. Achievable rate versus number of beams in the codebook in an outdoor LOS scenario under the receive SNR of 5 dB with different bits of quantization.

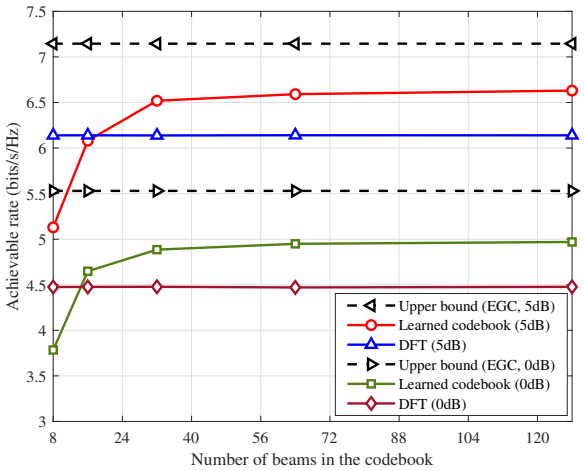


Fig. 4. The achievable rate versus number of beams in the codebook in an indoor NLOS scenario under the receive SNR of 0 and 5 dB.

phase shifters. It shows that with only 3-bit phase shifters, the learned codebook can still achieve more than 85% of the upper bound with 64 beams and 90% of the upper bound with 128 beams. This is very important and interesting for cases where the resolution of the analog phase shifters is limited.

Fig. 4 shows the evaluation result of the proposed solution in a NLOS scenario. This is more interesting and challenging than the LOS case. In a NLOS scenario, each user may have multiple rays reflecting off some scatters and reaching the base station. As depicted in Fig. 2, the learned codebook with the developed approach is capable of learning multi-lobe beams that capture the dominant rays and gather more energy compared to classical DFT codebooks. Fig. 4 depicts the performance of the proposed solution in this NLOS environment. **The learned codebook has almost the same performance or even outperforms a 64-beam DFT codebook with only 16 beams, which is one fourth in size**. With 64 beams, the learned codebook achieves almost 90% of the upper bound and noticeable gain compared to the 64-beam DFT codebook.

VII. Acknowledgment

This work is supported by the National Science Foundation under Grant No. 1923676.

VIII. Conclusions and Discussions

In this paper, we developed a machine learning framework for learning environment and hardware aware beamforming codebooks for large-scale MIMO systems. In the developed solution, the neural networks incorporate the hardware constraints and learn how to adapt the beam shapes based on the surrounding environment and user distribution. Simulation results confirmed the capability of the proposed solution in learning beam codebooks that are optimized in the size and beam patterns. This leads to promising gains in terms of the achievable rates compared to classical DFT codebooks. In the future, it is interesting to extend the proposed machine learning framework to address more complicated scenarios such as multi-stream and multi-user communications.

References

[1] R. W. Heath, N. Gonzlez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, April 2016.

[2] S. Hur, T. Kim, D. Love, J. Krogmeier, T. Thomas, and A. Ghosh, "Millimeter wave beamforming for wireless backhaul and access in small cell networks," *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4391–4403, Oct. 2013.

[3] D. Love, R. Heath, V. Lau, D. Gesbert, B. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems," *IEEE Journal on Selected Areas in Commun.*, vol. 26, no. 8, pp. 1341–1365, Oct. 2008.

[4] A. Alkhateeb, O. El Ayach, G. Leus, and R. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 831–846, Oct. 2014.

[5] J. Mo, B. L. Ng, S. Chang, P. Huang, M. N. Kulkarni, A. Alammouri, J. C. Zhang, J. Lee, and W. Choi, "Beam codebook design for 5G mmWave terminals," *IEEE Access*, vol. 7, pp. 98 387–98 404, 2019.

[6] A. Alkhateeb, J. Mo, N. Gonzalez-Prelcic, and R. Heath, "MIMO precoding and combining solutions for millimeter-wave systems," *IEEE Communications Magazine,*, vol. 52, no. 12, pp. 122–131, Dec. 2014.

[7] M. Alrabeiah and A. Alkhateeb, "Deep Learning for TDD and FDD Massive MIMO: Mapping Channels in Space and Frequency," *arXiv e-prints*, p. arXiv:1905.03761, May 2019.

[8] S. S. Haykin *et al.*, *Neural networks and learning machines/Simon Haykin*. New York: Prentice Hall,, 2009.

[9] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[10] F. Haslinger, *Complex Analysis: A Functional Analytic Approach*. Walter de Gruyter GmbH & Co KG, 2017.

[11] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep Complex Networks," *arXiv e-prints*, p. arXiv:1705.09792, May 2017.

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

[13] M. Alrabeiah and A. Alkhateeb, "Deep learning for mmwave beam and blockage prediction using Sub-6GHz channels," *submitted to IEEE Transactions on Communications, arXiv e-prints*, p. arXiv:1910.02900, Oct 2019.

[14] [Online]. Available: https://github.com/malrabeiah/learningCB