2021 5th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)
Military Institute of Science and Technology (MIST), Dhaka-1216, Bangladesh

Performance Analysis of Initialization Algorithms of Deep Neural Network Based Coordinated Beamforming System for mmWave

Dibaloke Chanda[1], Ali-Emam-Al-Badi[2], A K M Nazrul Islam[3]
Military Institute of Science & Technology (MIST), Dhaka - 1216, Bangladesh
dibaloke66@gmail.com[1]; aliemamalbadi@gmail.com[2]; nazrul@eece.mist.ac.bd[3]

***Abstract*—As mmWave has a wide range of applications, it has drawn a significant amount of attention in recent years. It has already been introduced in the next generation wireless communication system. In practice, it shows some shortcomings and most of these are eliminated by introducing beamforming which utilizes the spatial diversity enabled by Massive MIMO. Still, there are a few challenges in designing an efficient system for highly mobile users and making sure proper coverage and reliability. In this research, a machine learning-based coordinated beamforming technique has been explored that supports highly mobile applications in mmWave systems with massive antenna arrays. The optimization of the deep learning model itself can increase the system performance as well as reduce the computational time complexity. The purpose of this work was to optimize the deep learning model and recommend proper initialization method to maximize the system performance. We found that for Xavier normal initialization algorithm the effective achievable rate is the highest for the least amount of data.**

***Index Terms*—MIMO, Beamforming, Initialization, mmWave ,**

I. Introduction

Within the last few years, millimeter wave (mmWave) communication has gained a lot of attention. Next generation wireless systems have already adopted mmWave technology to make sure high data speeds provided by its vast accessible bandwidth. The shortcomings of mmWave communication system are mitigated with technology like beamforming which utilizes spatial diversity enabled by Massive MIMO. But there are a few challenges when designing a highly mobile mmWave system. Firstly, due to the high mobility of the user and dense deployment there need to be frequent handover between base stations otherwise it might endanger the reliability of connection and result in poor user experience, specially for vehicular communication. This can also result in latency and jitter which is not acceptable. Secondly, for large antenna arrays, constructing adequate beamforming vectors with standard methods needs a considerable training overhead with traditional algorithms which poses a question if mmWave systems are suitable for highly mobile users.Some prior work has already been done to answer this question with help from machine learning and coordinated beamforming, combining them to form a solution. Coordinated beamforming uses a central hub to connect several base stations which simultaneously serves a user to increase reliability and counteract against ill-conditioned channels and deep learning based algorithms construct proper beamforming vectors to reduce the system's training overhead and complexity. The optimization of the deep learning model itself can increase the system performance as well as reduce the computational time complexity. In this research work, this aspect of deep learning based coordinated beamforming system is explored and thoroughly analyzed.

II. Literature Review

Some prior work explored various aspects of mmWave systems like signal outage and coverage. Maamari et al. [1] studied the base station collaboration in the downlink of dense mmWave heterogeneous network to reduce signal outage and combat blocking. Coverage probabilities were calculated for a typical user, taking into account base station directionality, obstruction, interference, and different fading distributions. They concluded coverage with coordinated beamforming is far superior than beamforming, especially in dense mmWave networks.

In an urban micro cell open square scenario in downtown Brooklyn, New York, on the NYU campus, extensive measurement was done in a proper practical scenario. Ten random receiver locations at the pedestrian level (1.4 meters) and ten random transmitter locations at lamppost level (4.0 meters) produced 36 unique transmitter-receiver (TX-RX) combinations for the measurements [2].

The authors discovered that one to five base stations serving a single RX location improve coverage significantly when compared to all possible beamformed RX antenna pointing angles. Several research tried to integrate deep learning algorithms with beamforming. In a paper by Guo et al. [3] past channel state information (CSI) was used to create a machine learning prediction model with LSTM to efficiently forecast the future channel. The authors demonstrated that the suggested LSTM can properly forecast the vehicle user's channel and achieve a sufficient transmission rate while requiring less pilot overhead than a standard beam training technique. Wang et al. [4] explored how to construct a compressive beam alignment in mmWave vehicle systems using deep learning. This particular research leveraged the sparsity of the mmWave channel to develop a convolutional neural network. In a similar research work, Alkhateeb et al. [5], a coordinated beamforming system was developed, in which a deep learning model learned how to predict beamforming vectors from the receiver of distributed BSs using only omni or quasi-omni beam patterns. As a prerequisite a clear understanding of the mmWave system, channel model and basic signal processing techniques is required to understand how machine learning algorithms can be applied to mmWave systems as well as understanding how MIMO is used mmWave [6]–[8]. The authors in these papers introduce these fundamental concepts starting from basic digital communication theory and how they evolve into mmWave technology [9]. To optimize the machine learning model, a basic knowledge about various optimization algorithms is required as well as their mathematical formulation and intuition about how they work. The authors mainly focused on developing the intuition about these optimization algorithms rather than going into rigorous formulation because these optimization algorithms are based on empirical data and heuristic techniques [10]–[13]. The authors go in depth about these optimization algorithms and how they works starting from the basic mathematical formulation. In addition how to overcome some limitations of these optimization algorithms provides a intuition about their mathematical reasoning [14]. The basic of cost function and initialization algorithm is also a perquisite for understanding, how optimizing these aspects increases the system performance [15].

III. Dataset

The data set used here was obtained from previous research works which focused on a DeepMIMO data set generation framework. The authors of the paper used RemcomWireless InSite to get accurate ray-tracing data to generate the channel matrix of MIMO communication system. The data set generation framework provides the facility to change various parameters of the model like number of active BSs, number of active users, number of BS antennas, antenna spacing, system bandwidth, number of OFDM sub-carriers, OFDM sampling factor, OFDM limit, number of paths etc. The dataset was generated based on the framework for an outdoor environment [16]. The dataset is open source and a couple of scenarios was provided, from which O1 scenario was used in this research [17].

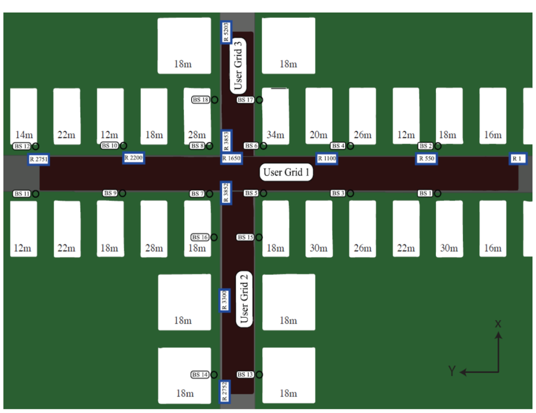

Fig. 1. O1 scenario(top view) [17]

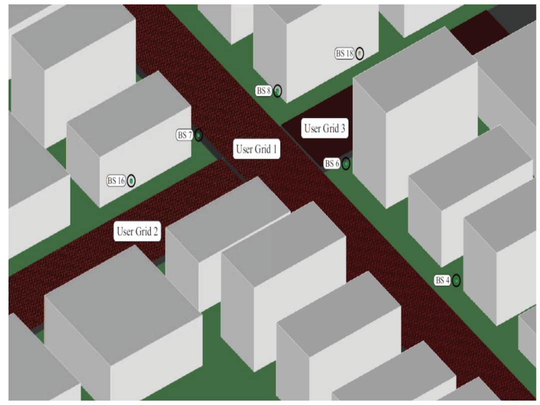

Fig. 2. O1 scenario (bird's eye view) [17]

The 'O1' ray-tracing scenario is an outdoor scenario of two streets and one intersection with the top-view and the bird-view shown in fig. 1 and fig. 2 . The main street (the horizontal one) is 600 m long and 40 m wide, and the second street (the vertical one) is 440 m long and 40 m wide. The operating frequency is 60 GHz.

IV. Methodologies

A. Coordinated Beamforming

In coordinated beamforming, a single user is served by multiple base stations as shown in fig. 3. The base stations are connected to a central hub where processing occurs. As mmWave system works on high frequency range, the operating wavelength shrinks. As a result, the antenna size in mmWave communication is significantly small compared to the earlier generation of wireless technology.

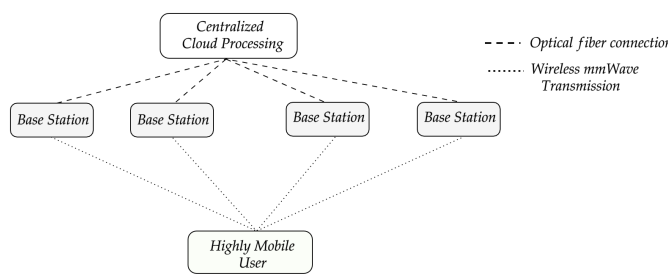

Fig. 3. Coordinated beamforming architecture

The base stations are connected to central hub via optical fiber to make sure no interruption of communication takes place between the base stations and the central hub. The signals coming from all these base station adds in such a way that the user will always get a reliable connection. The training overhead is really large as all these base stations need to work in synchronization.

B. System Model

The ML(Machine Learning) model is shown in the fig. 4 .The pilot sequence received by omni directional antenna is fed to a dense neural network.

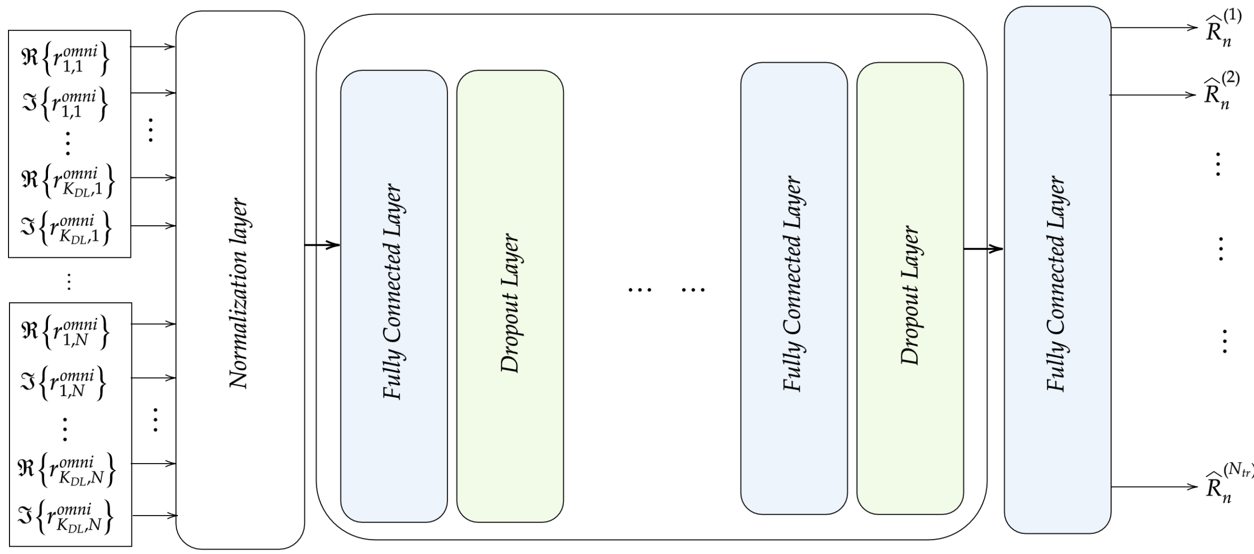

Fig. 4. System model

The neural network consists of input normalization layer, followed by a fully connected and dropout layer. The fully connected and dropout layer is repeated a couple of times depending on how deep the neural network is. In this paper, that stage was repeated 4 times. The received signals by the base stations are represented by $r_{k,n}^{\text{omni}}$, where the superscript signifies omnidirectional antennas were used to receive the pilot sequences and k represents k^{th} OFDM subcarrier, n denotes the sequence was received by n^{th} antenna of the base station. There are total N base stations in the system.

The output of the neural network $\hat{R_n}$ are the beamforming vectors which determines the direction of the beam. The channel is considered a block fading which stays constant over the channel coherence time T_c.

V. Initializer Algorithms

If the weights of a ML model are initialized with all zeros, it leads to a problem called symmetry breaking problem. When this happens all weights will get the same update. This reduces the degrees of freedom of the weight updates. Same thing will happen if all the weights are initialized to the same value. This kind of initialization halts the learning process completely and results in a poor performance. Initializing the weights to a really small value results in vanishing gradient problem. The key point is that the calculated partial derivatives are used to compute the gradient. Depending on the value of the gradients defines how much the network will learn during the training, if the gradients are very small or zero, no training can take place, leading to poor predictive performance. For a shallow network with only a few layers that use these activation's, this isn't a big problem. However, more layers can cause the gradient to be too small for training to work effectively. On the other end of the spectrum, if the weights are initialized to a very large value, it can result in an issue known as exploding gradient. The exploding gradient problem is caused by the same reason that caused the vanishing gradient problem. Large erroneous gradients build and result in very large updates to the neural network model weights during training, which is known as exploding gradients.

A couple of key points to avoid the above mentioned issues:-

1) Weights should not be initialised with zero or the same value.
2) It's possible to break the symmetry by initializing the weight randomly and keep the bias constant.
3) Weights should not be initialised with large value.
4) There should be some kind of mechanism like gradient clipping to reduce the gradient if the gradients start to get really large.

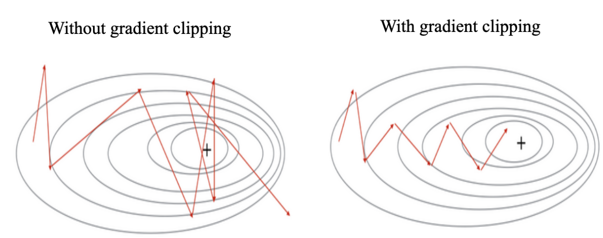

Fig. 5. Effect of gradient clipping

As seen from the fig. 5, gradient clipping can reduce the problem with exploding gradient and can find a local minima.

Another option is to use values from a random normal distribution to start the weights. It is possible to establish the weight with a defined range by setting the mean and standard deviation using the normal distribution. Setting the mean to zero and the standard deviation to one is an example.

This approach randomizes all weights, which is much preferable than setting a static value.Due to the random nature

, the model can have a higher chance of starting near a local minima, most importantly this approach erases the symmetry problem.Also, the normal distribution have some nice mathematical properties which makes it an ideal choice.Its also possible to sample from other gaussian distribution with mean not at zero and variance different from one . But that can make the model unstable as weights can be set at really large values which can result in exploding gradient problem.So, if that approach is used there has to be a normalization layer which restrict the values from 0 to 1.That way, it is possible to avoid exploding gradient problem. There are many distributions, but in practice the normal distribution is used the most.Fig. 6 shows a normal distribution where numbers are randomly sampled.

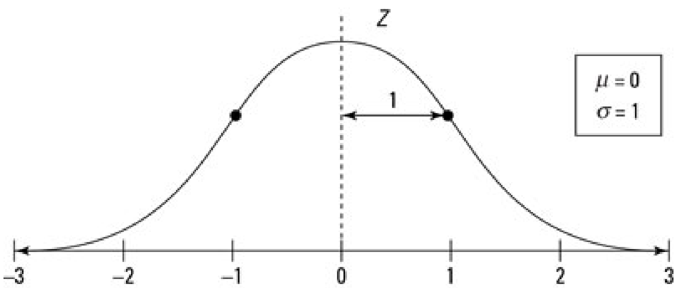

Fig. 6. Normal distribution

A better way to initialize the parameters in a neural network is to control the variance of the output. Xavier initialization [15] is the technique that initializes the weights in a way that the output produced by the neurons all follow the same distribution.The idea behind the Xavier initialization can be summarized into two key points : -

1) The mean of the activations should be zero.
2) The variance of the activations should stay the same across every layer.

Two variants of Xavier initialization are Xavier uniform initialization and Xavier normal initialization.

Although its a uniform distribution it's performance is relatively good compared to random uniform distribution.

$$W_{ij} \sim U\left[-\frac{1}{\sqrt{k}}, \frac{1}{\sqrt{k}}\right] \tag{1}$$

Eq.1 represents the Xavier uniform distribution where U is the uniform distribution in the interval $(\frac{1}{\sqrt{k}}, \frac{1}{\sqrt{k}})$ and k is the size of the previous layer (the number of columns of W).The following equation represents a more generalized version,which is called normalized initialization.

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{k_j + k_{j+1}}}, \frac{\sqrt{6}}{\sqrt{k_j + k_{j+1}}}\right] \tag{2}$$

Therefore, using this approach, samples are drawn from a normal distribution, where the normal distribution is described by the above equations.

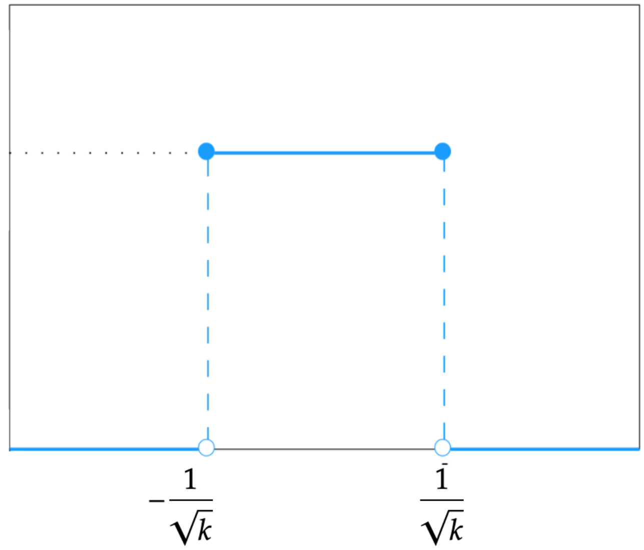

Fig. 7. Xavier uniform distribution

The following equation shows xavier random distribution meaning samples are drawn from a normal gaussian distribution, where the normal distribution is described by the above equation.

$$W \sim \mathcal{N}\left(0, \frac{2}{k^{[l-1]} + k^{[l]}}\right) \tag{3}$$

Xavier almost always works better compared to other initialization methods. But depending on the particular problem other initialization methods can be chosen.

VI. Simulation Results

The simulation was done to observe the effect of how different initialization affects the system performance. The main goal is to get the maximum effective achievable rate with less percentage of the dataset.Genie-Aided performance is not possible in practical scenarios, as it would require the perfect channel knowledge, but is used as a benchmark. ML model parameters used during the simulation are shown in Table I.

TABLE I
Parameters of Initialization Methods Simulation

Parameter Name	Value
Number of hidden layers	8
Input shape	(724, 256)
Output shape	(724, 2048)
Activation function	reLU
Number of epoch	10
Batch size	100

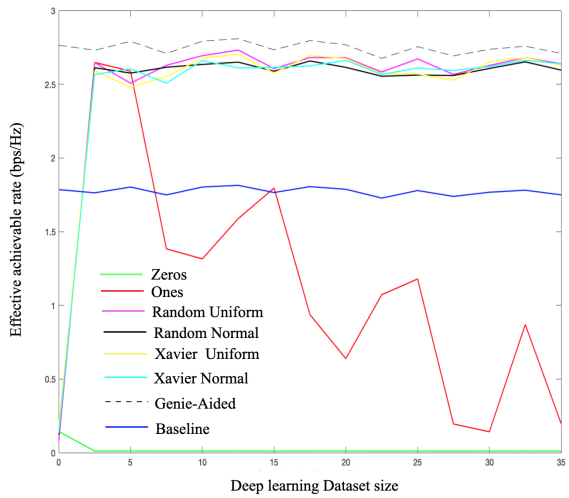

Fig. 8. Initializer comparison plot

Simulation result for different initialization and the comparison between different initializations is illustrated in this section. The output of the simulation reflects the intuition developed in the preceding sections. The results of deep learning coordinated beamforming will be compared to baseline coordinated beamforming, which is based on traditional code books.

Initializing weights to zero results in really poor performance. The effective achievable rate of deep learning based coordinated beamforming is nowhere near the baseline coordinated beamforming irrespective of the data set size. This happens due to the symmetry breaking problem.

In case of constant weight initialization, all the weights are given constant value. This constant value can be any random number, in this simulation the constant number was taken as 1. From fig. 8 it's apparent that this initialization method has the same problem as zeros initialization method which is reflected in the plot. But for really small percentage of the data set this actually performs better than baseline coordinated beamforming. But then its start decreasing as the percentage of dataset increases.

Compared to constant weight initialization, initializing weights with random numbers increase the system performance by a significant amount. There is a slight increase in performance for random normal initialization compared to random uniform initialization. This randomness of initialization gives a large degree of freedom for the weights and biases of the system. As for random normal initialization the performance stays constant for any percentage of the data set size.

Xavier's initialization algorithms shows the best performance compared to other initialization methods especially xavier normal initialization. They reach the peak value of effective achievable rate with data set size as low as 5%. We know that Xavier intialization tries to make the variance of the outputs of a layer to be equal to the variance of its inputs. In case of Xavier uniform initialization it combines the benefit of uniform initialization and Xavier's method . Though it dips by a small amount at 5% data set. For this reason, it is inferior to Xavier normal distribution . But it can be concluded that it is better than baseline coordinated beamforming and close to the genie-aided beamforming.Therefore, Xavier normal initialization will result in the most optimised model and will reduce the computational complexity as well as produce the best performance. The reason behind Xavier initialization method to perform this well is it's inherent nature to keep the variance same. The variance remains the same after each layer is passed, so when doing the feed forward algorithm after each layer the variance will remain unchanged.

VII. Future Work

This research work only explored the performance of the system for different initialization algorithms. By changing the number of neurons in each layer, number of epochs, learning rate further investigation can be done to see the effects of the changes on the system performance. Also, the analysis has been done for line of sight (LOS) environment, it can also be done under non line of sight (NLOS) environment. Another scope for future work is trying out different machine learning architectures like convolutional neural networks, recurrent networks and analyze how the system performance changes.

VIII. Conclusion

In this paper, we explored a deep learning based approach to predict beamforming vectors in mmWave system, specifically for coordinated beamforming. By analyzing different initialization algorithms, it was found that Xavier normal initializer algorithm provides the best effective achievable rate for the least percentage of data.It is also apparent, its performance is better than traditional coordinated beamforming. Xavier Uniform initializer also performs relatively well, but zero initialization and constant initialization results in really poor performance.

References

[1] D. Maamari, N. Devroye, and D. Tuninetti, "Coverage in mmwave cellular networks with base station co-operation," *IEEE transactions on Wireless Communications*, vol. 15, no. 4, pp. 2981–2994, 2016.

[2] G. R. MacCartney, T. S. Rappaport, and A. Ghosh, "Base station diversity propagation measurements at 73 ghz millimeter-wave for 5g coordinated multipoint (comp) analysis," in *2017 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2017, pp. 1–7.

[3] Y. Guo, Z. Wang, M. Li, and Q. Liu, "Machine learning based mmwave channel tracking in vehicular scenario," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[4] Y. Wang, N. J. Myers, N. González-Prelcic, and R. W. Heath Jr, "Deep learning-based compressive beam alignment in mmwave vehicular systems," *arXiv preprint arXiv:2103.00125*, 2021.

[5] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[6] R. W. Heath, N. Gonzalez-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, 2016.

[7] T. S. Rappaport, "Wireless communications–principles and practice, (the book end)," *Microwave Journal*, vol. 45, no. 12, pp. 128–129, 2002.

[8] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE communications magazine*, vol. 49, no. 6, pp. 101–107, 2011.
[9] A. M. Sayeed, T. Sivanadyan, K. Liu, and S. Haykin, "Wireless communication and sensing in multipath environments using multi-antenna transceivers," in *Handbook on Array Processing and Sensor Networks*. Wiley Online Library, 2010, pp. 115–170.
[10] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, 2012.
[11] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A survey of optimization methods from a machine learning perspective," 2019.
[12] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.
[13] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," *arXiv preprint arXiv:1406.2572*, 2014.
[14] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points—online stochastic gradient for tensor decomposition," in *Conference on learning theory*. PMLR, 2015, pp. 797–842.
[15] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.
[16] A. Alkhateeb, "Deepmimo: A generic deep learning dataset for millimeter wave and massive mimo applications," *arXiv preprint arXiv:1902.06435*, 2019.
[17] "Available at," http://deepmimo.net/ray_tracing.html, accessed: 2020-05-16.