Report on

BEAM PREDICTION USING DEEP LEARNING

Submitted in fulfilment of

Study Project (EEE F266)

by

Sagar Kaushik (2017B5A30912P)

under the guidance of

Dr. Sandeep Joshi

May 2021

Abstract

Much of commercially available spectrum to consumers is in lower frequency bands, and therefore higher frequency like Millimetre-Waves (mmWaves), that have higher bandwidths and data rates, are thought to be one of the next major developments in wireless communication. But mmWaves have two major drawbacks, there is a large training overhead for adjusting beamforming vectors and they are very sensitive to blockages. On the other hand, conventional waves (sub-6GHz) have a low training overhead and are not blocked easily. Future wireless networks will likely be dual-band — operating at both sub-6GHz and mmWave band, and it can be proven that under certain conditions, there exist mapping functions that can predict the optimal mmWave beam from the sub-6GHz channel. This helps in saving training time and resources. Since these functions are hard to categorize analytically, a deep neural network can be used for the task such that there is no beam training overhead. Results show that deep learning models can predict the appropriate mmWave beam in with more than 90% accuracy.

<u>**Index**</u>

INTRODUCTION

It is very difficult to achieve and maintain the *claimed* high data rate gains of mmWave communications that have high mobility and reliability constraints because of i) large training overhead of adjusting beamforming vectors and ii) sensitivity of mmWaves to blockages. These problems get tougher with higher frequencies and large antenna arrays. Sub-6GHz waves on the other hand have a low training overhead and are less susceptible to blockages. We can use the spatial correlation between these two bands to remedy the problems with mmWaves to a great extent.

Beam management processes happen to establish and maintain connection between the base station (BS) and user device as follows:

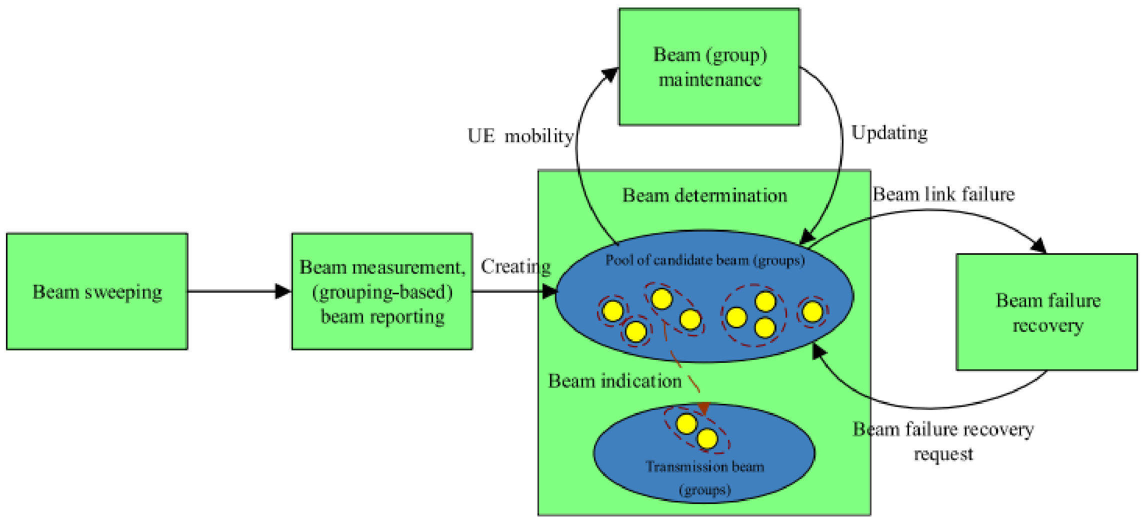

Beam (group) maintenance
UE mobility
Updating
Beam determination
Beam link failure
Beam sweeping
Beam measurement, (grouping-based) beam reporting
Creating
Pool of candidate beam (groups)
Beam failure recovery
Beam indication
Beam failure recovery request
Transmission beam (groups)

Figure 1. Beam management processes

This is followed by beamforming. Beamforming (BF) is focusing of signal in some direction(s) so that it reaches user equipment in a focussed manner (UE). It is of three types:

- Analog BF: Single Radio Frequency (RF) chain for all antennas, therefore, works in one direction at a time and employs phase shifters.
- Digital BF: A separate RF chain for each antenna element. Transmit power is divided between all, so it is low.
- Hybrid BF: Equivalent to parallel analog beams transmitting/receiving in ‘K’ directions because of ‘K’ RF chains.

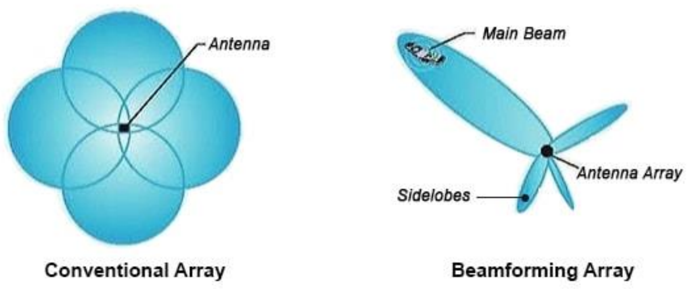


Figure 2. Conventional array creates quasi-omnidirectional beams and beamforming array creates directional beam lobes

Due to higher power requirements for higher frequencies required for a unique RF chain for each antenna element in case of digital BF, mmWaves are expected to use either hybrid or analog BF.

Consider a system where two antenna arrays belonging to the mmWave (analog) and sub-6GHz (digital) transceivers at base station (BS) are co-located and assume that the uplink signalling happens at the sub-6GHz band while the downlink data transmission occurs at the mmWave band. Mobile station has one antenna for sub-6GHz and one for mmWave.

For such a system, the uplink received k*th* subcarrier (k = 1,...K) signal at the BS is

$$\mathbf{y}_{\text{sub-6}}[k] = \mathbf{h}_{\text{sub-6}}[k] s_{\text{p}}[k] + \mathbf{n}_{\text{sub-6}}[k],$$

where $h_{\text{sub-6}}[k]$ is the uplink channel vector and $s_p[k]$ represents the uplink pilot signal. The vector $n_{\text{sub-6}}[k] \sim N_C(0, \sigma^2 I)$ is the receive noise at the BS sub-6GHz array.

For downlink transmission,

$$y_{\text{mmW}}[\bar{k}] = \mathbf{h}^T_{\text{mmW}}[\bar{k}]\mathbf{f} s_{\text{d}} + n_{\text{mmW}}[\bar{k}],$$

where $h_{\text{mmW}}[k]$ is the downlink channel vector, f is the downlink beam forming vector and $s_d[k]$ represents the downlink pilot signal. The vector $n_{\text{mmW}}[k]$ is the receive noise at the mobile station mmWave array.

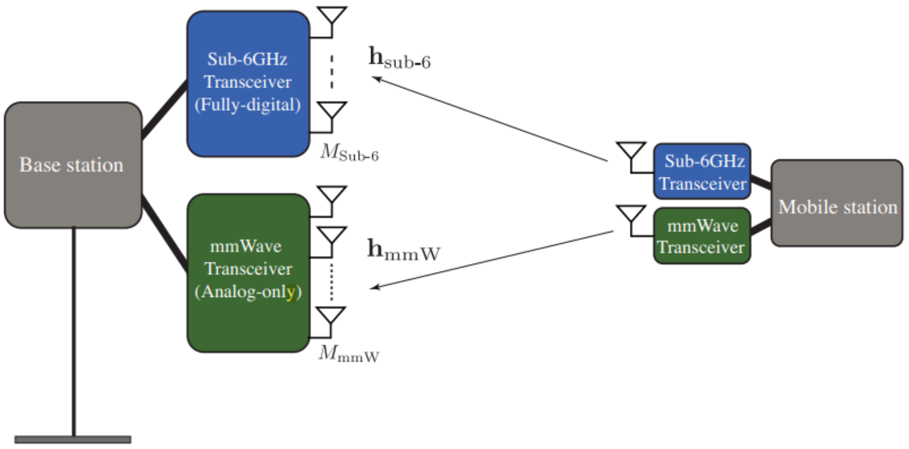

Sub-6GHz Transceiver (Fully-digital)
$\mathbf{h}_{\text{sub-6}}$
$M_{\text{Sub-6}}$
Base station
mmWave Transceiver (Analog-only)
$\mathbf{h}_{\text{mmW}}$
M_{mmW}
Sub-6GHz Transceiver
mmWave Transceiver
Mobile station

Figure 3. System schematic

Due to the hardware constraints the beamforming vectors (**f**) are normally selected from quantized codebooks.

For the channel, a physical/geometrical model is adopted for its ability to catch the physical characteristics of signal propagation. For mmWave (and similarly for sub-6 GHz):

$$\mathbf{h}_{\text{mmW}}[k] = \sum_{d=0}^{D-1} \sum_{\ell=1}^{L} \alpha_\ell e^{-j\frac{2\pi k}{K}d} p\left(dT_{\text{S}} - \tau_\ell\right) \mathbf{a}\left(\theta_\ell, \phi_\ell\right),$$

where L is number of channel paths, α_l, τ_l, θ_l, φ_l are the path gains (including the path-loss), the delay, the azimuth angle of arrival (AoA), and elevation AoA, respectively, of the *l*th channel path. T_S represents the sampling time while D denotes the cyclic prefix length (assuming that the maximum delay is less than DT_S).

PROBLEM DEFINITION

For the above, described model, the downlink achievable rate is:

$$R\left(\left\{\mathbf{h}_{\mathrm{mmW}}[\bar{k}]\right\}, \mathbf{f}\right)=\sum_{\bar{k}=1}^{\bar{K}} \log_2\left(1+\mathsf{SNR}\left|\mathbf{h}_{\mathrm{mmW}}[\bar{k}]^T \mathbf{f}\right|^2\right),$$

The optimal beamforming vector $\mathbf{f}^*$ that maximizes R is given by exhaustive search:

$$\mathbf{f}^{\star}=\underset{\mathbf{f} \in \mathcal{F}}{\arg\max}\, R\left(\left\{\mathbf{h}_{\mathrm{mmW}}[\bar{k}]\right\}, \mathbf{f}\right),$$

yielding the optimal rate R^*. But performing this search requires either estimating the mmWave channel h_{mmW} or an online exhaustive beam training, and both have large training overhead.

We can reduce this training overhead by using sub-6GHz channel information to predict and decide the optimal BF vector.

DEEP LEARNING APPROACH

In deep learning (DL), some features are chosen and then fed as input to a neural network, which then tries to map that to the given output. Depending on the prediction, the weights of neurons are changed so as the model learns to be more accurate, this is called training. Then the model is tested on new data. These models can learn and approximate non-trivial functions. We can use that for prediction of the optimal mmWave beams directly from the knowledge of the sub-6GHz channels with a very high accuracy.

Deep learning based operation requires no learning overhead in terms of the system time-frequency resources because mmWave beam training will typically be performed anyway, in systems that do not use machine learning, to find out the best beamforming direction. The only practical issue with this approach is that it was assumed that the mapping from the user positions to the sub-6GHz channels is bijective (one-to-one), and therefore the sub-6GHz channels can be used to predict the optimal mmWave beams, but in reality, a few factors can add some probabilistic error to this beam prediction such as the measurement noise, the phase noise, and the dynamic scatterers in the environment. And these factors can make the position-to-channel mapping not perfectly bijective or create sub-6GHz channels that were not experienced before by the neural networks.

DEEPMIMO DATASET

The DL model was trained using the DeepMIMO dataset in which channels are constructed from accurate ray-tracing data obtained from the simulator, Remcom Wireless InSite. Raytracing scenarios, for different geographical distributions both outdoor and indoors, are obtained by giving various number of base stations & users for various frequencies (denoted by R).

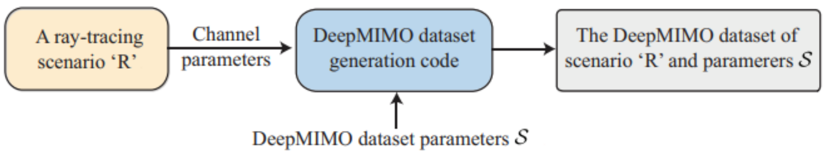

A ray-tracing scenario 'R'
Channel parameters
DeepMIMO dataset generation code
The DeepMIMO dataset of scenario 'R' and paramerers $\mathcal{S}$
DeepMIMO dataset parameters $\mathcal{S}$

Figure 4. Dataset generation flowchart

Parameters like active BSs, active users, number of BS antennas, antenna spacing, bandwidth, OFDM sampling factor & limit and number of channel paths are also used (denoted by S). After choosing 'R' and 'S' and feeding those to the MATLAB generation code, we get a dataset that we can use for DL applications.

For this problem, O1 scenario (outdoor 1) was taken for the two frequencies, 3.5GHz (sub-6GHz) and 28GHz (high frequency mmWave). It has 18 total base stations. User grid 1 (R1-R2751) has 181 users in each row. Similarly, user grid 2 (R2752 to R3852) also has 181 users at every row. User grid 3 (R3853 to R5203), the last one, has 361 users in each row.

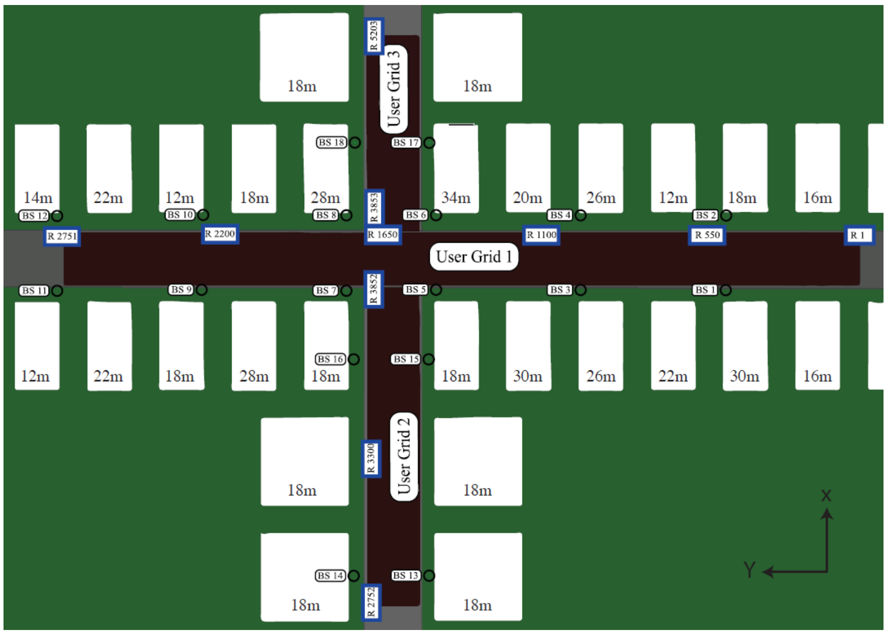
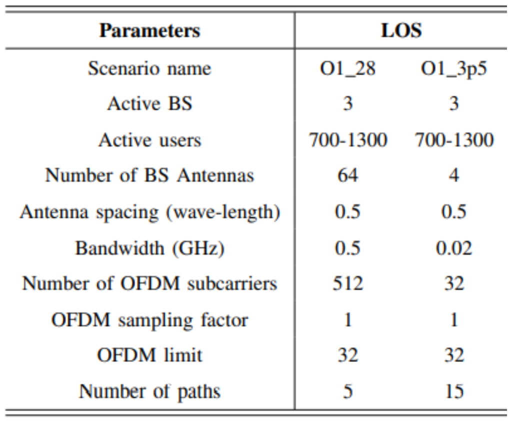


Figure 5. O1 scenario grid layout

We consider grid 1 for our dataset, active users from R700 to R1300 and active BS 3. BS is assumed to be equipped with 64 antennas for 28GHz wave and 4 antennas for 3.5GHz one. Bandwidths are taken as 0.5GHz and 0.02GHz for mmWave and sub-6GHz, respectively. Antenna spacing is set as 0.5 of the wavelengths, OFDM sampling as 1 and OFDM limit as 32, in both the cases. Number of OFDM carriers are taken to be 512 and 32, while number of channel paths 5 and 15 for mmWave and sub-6GHz, respectively. Only line of sight (LOS) ray paths are considered.

Table 1. Parameter values used for dataset generation



Parameters	LOS	
Scenario name	O1_28	O1_3p5
Active BS	3	3
Active users	700-1300	700-1300
Number of BS Antennas	64	4
Antenna spacing (wave-length)	0.5	0.5
Bandwidth (GHz)	0.5	0.02
Number of OFDM subcarriers	512	32
OFDM sampling factor	1	1
OFDM limit	32	32
Number of paths	5	15

DEEP LEARNING MODEL

Since the beamforming vectors for mmWave are usually selected through exhaustive search of a quantized codebook, we can use those finite number of vectors to represent different classes and treat this problem as a basic multi-class classification. Classification is a very common problem that can be easily and successfully solved using DL. In this model, for some given input, the neural network will try to figure out the appropriate mmWave BF class (output), and it will learn by adjusting the neuron weights based on the results. After training is done, new unencountered data will be given to the network for validation. If the model works well, it can be used to predict mmWave beams in real time, saving the search time.

Authors' Network Architecture

A multi-layer perceptron (MLP) network, that consists of 6 stacks. Each of the first 5 stacks have 2048 neurons, followed by ReLU activation function and then a drop out layer (all have 0.1 probability). Last stack has 64 neurons, one for each class, followed by SoftMax function for finding probability distribution of each class. Sub-6GHz channels are taken as inputs and Stochastic Gradient Descent is used as optimizer.

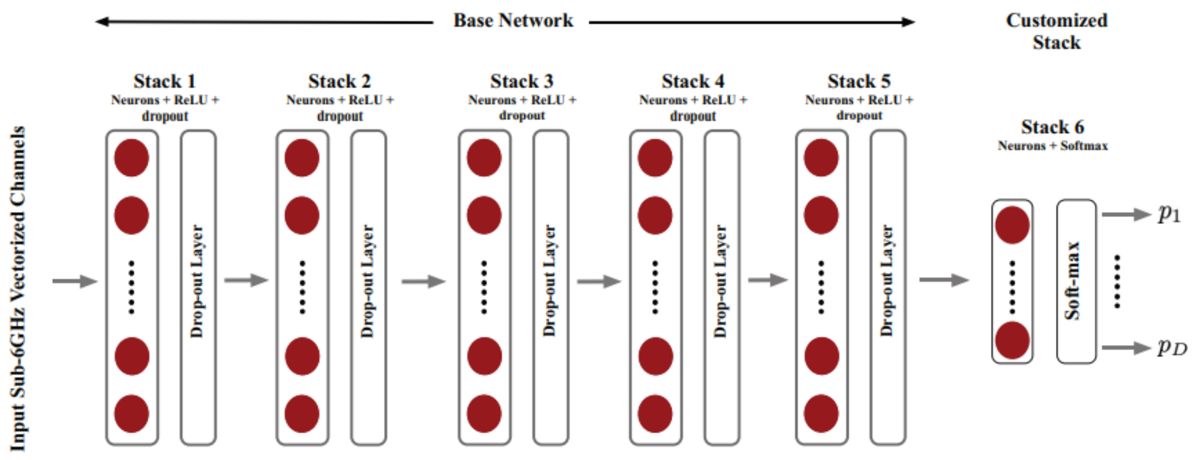


Figure 6. Authors' network

Changes Made

- While the original network only took sub-6GHz channels as inputs, I added user location (X and Y coordinates) as two more features.
- Changed activation function from ReLU to Mish for each layer to eliminate the possibility of "dying ReLU" for any neuron.
- Swapped out Stochastic Gradient Descent optimizer with Adam optimizer, which is faster, though computationally more expensive.
- Increased dropout from 0.1 to 0.3 since model was overfitting.
- Inserted a learning rate scheduler so that learning step decreases as accuracy stops increasing.
- Also added EarlyStop callback so that training stops when validation loss starts increasing or if it plateaus, to save training time.

Number of neurons and number of layers were kept the same. Number of epochs = 100.

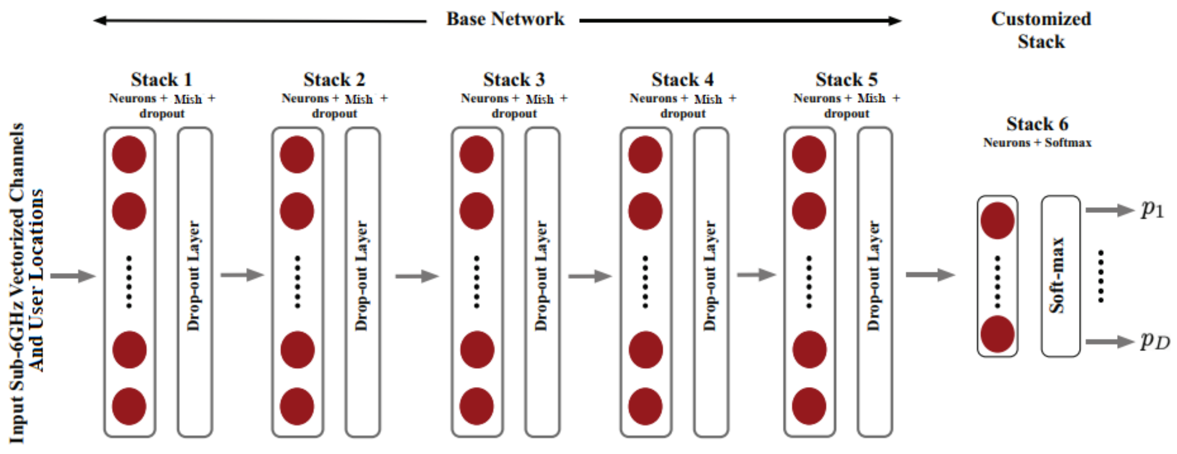
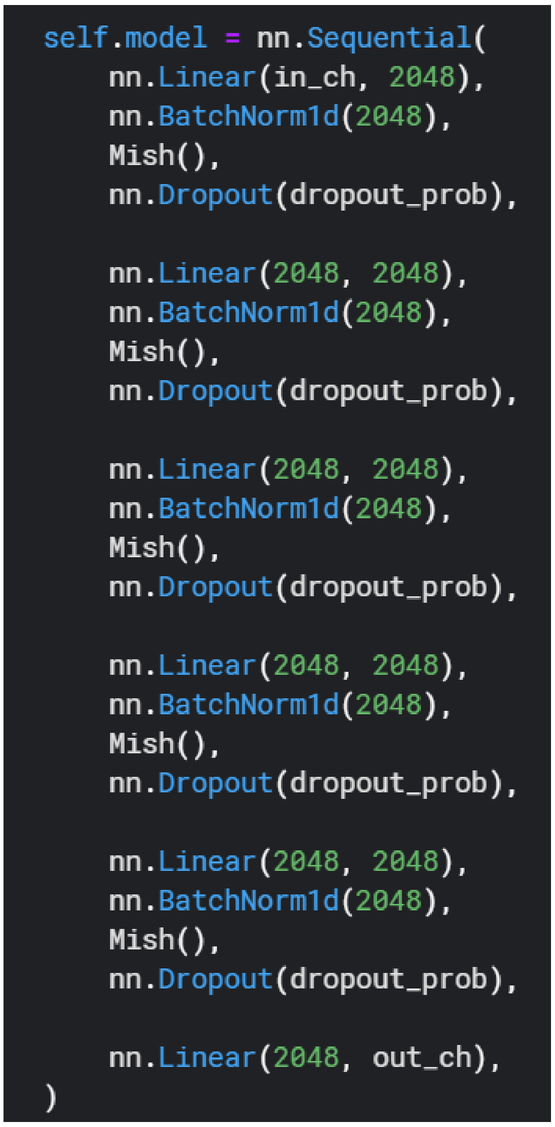
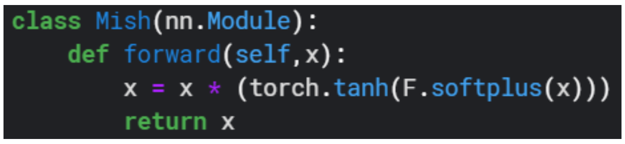


Figure 7. Changed network



```
self.model = nn.Sequential(
    nn.Linear(in_ch, 2048),
    nn.BatchNorm1d(2048),
    Mish(),
    nn.Dropout(dropout_prob),

    nn.Linear(2048, 2048),
    nn.BatchNorm1d(2048),
    Mish(),
    nn.Dropout(dropout_prob),

    nn.Linear(2048, 2048),
    nn.BatchNorm1d(2048),
    Mish(),
    nn.Dropout(dropout_prob),

    nn.Linear(2048, 2048),
    nn.BatchNorm1d(2048),
    Mish(),
    nn.Dropout(dropout_prob),

    nn.Linear(2048, 2048),
    nn.BatchNorm1d(2048),
    Mish(),
    nn.Dropout(dropout_prob),

    nn.Linear(2048, out_ch),
)
```

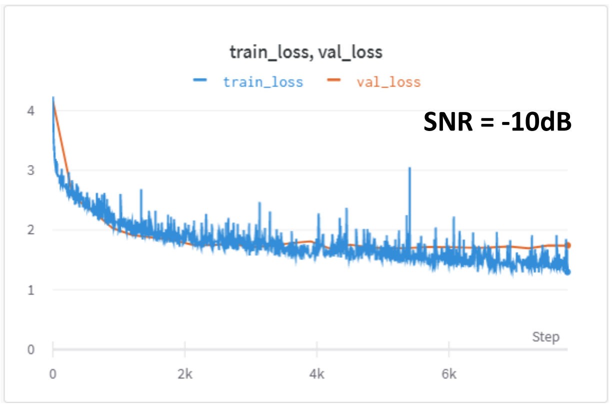


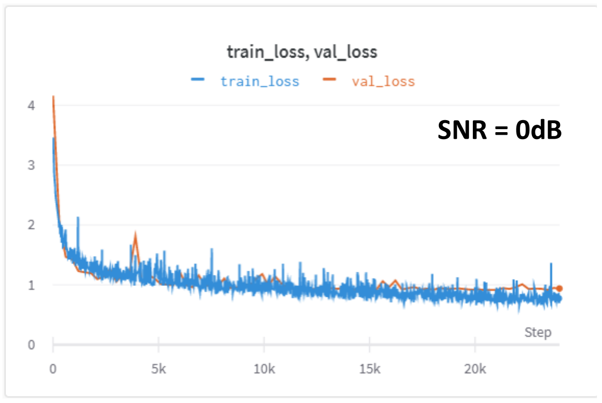
```
class Mish(nn.Module):
    def forward(self,x):
        x = x * (torch.tanh(F.softplus(x)))
        return x
```

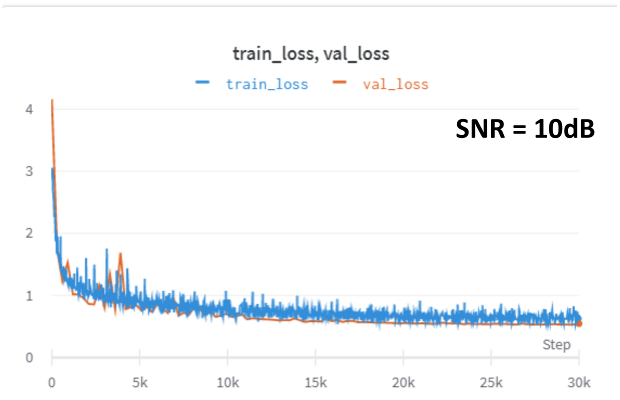
Figure 8. Model code and Mish activation function

RESULTS

Losses







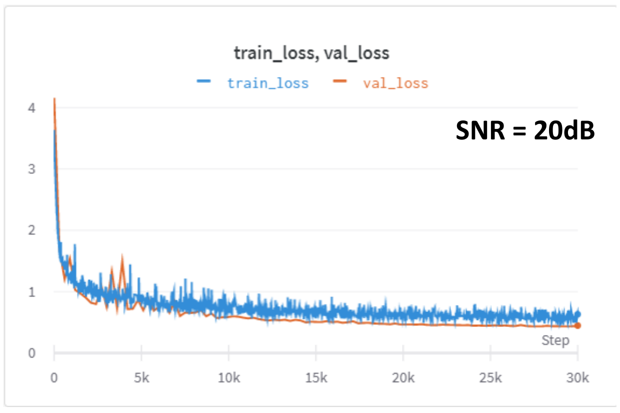
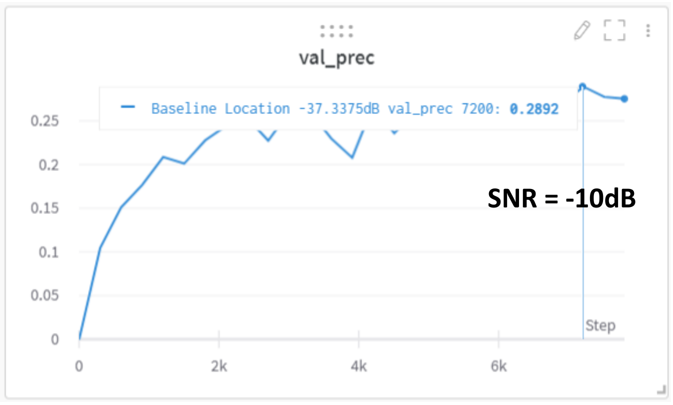
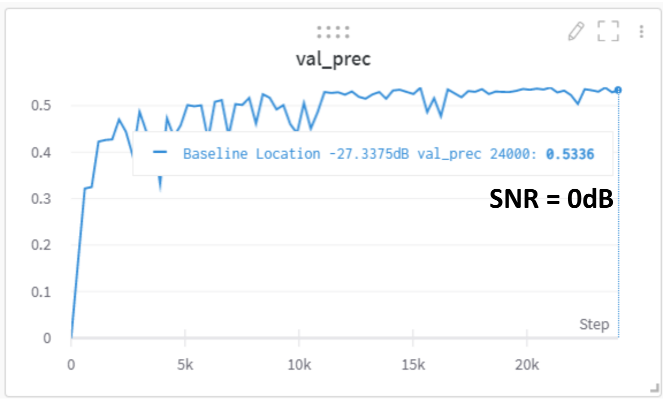
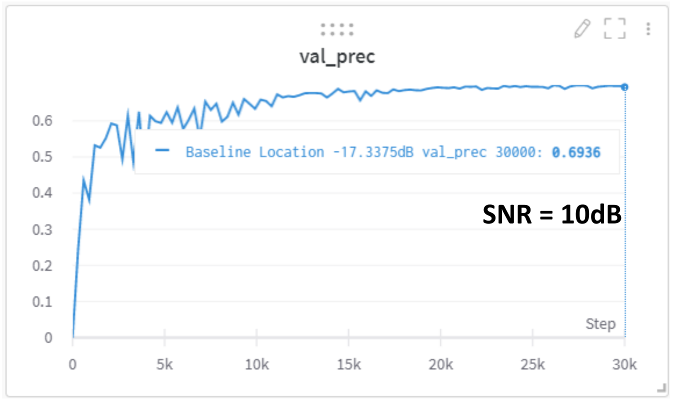


Figure 9. Losses for four SNRs

Precision







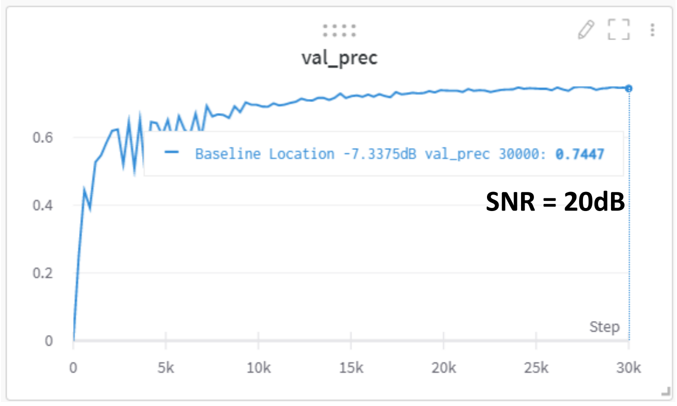
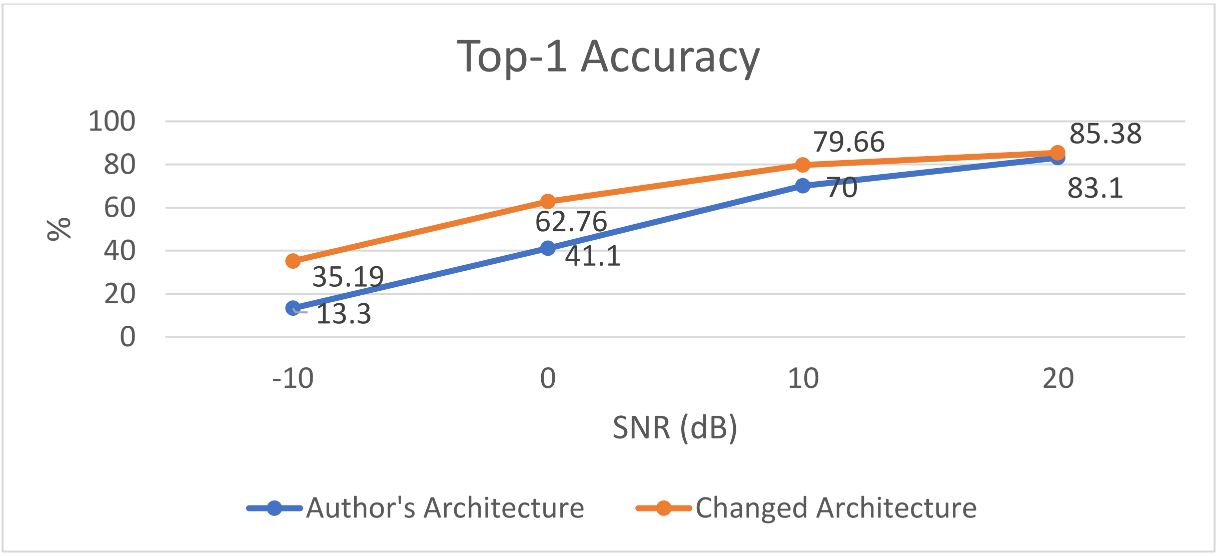
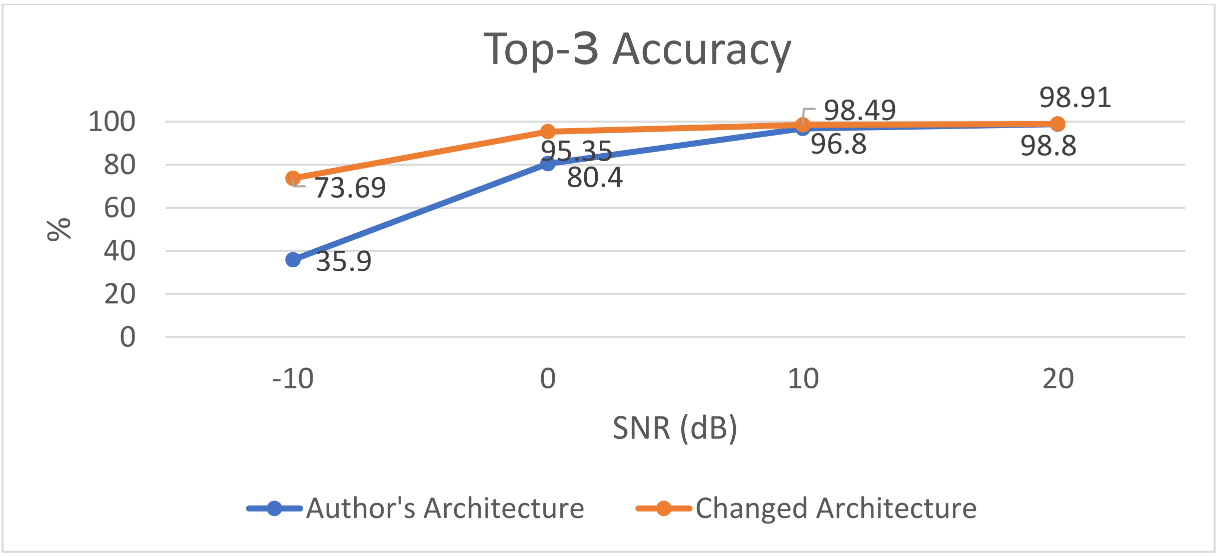


Figure 10. Precision for four SNRs

Accuracies

Accuracy (%)	Top-1		Top-3	
SNR (dB)	Author's Network	Changed Network	Author's Network	Changed Network
-10	13.30	35.19	35.9	73.69
0	41.10	62.76	80.4	95.35
10	70.00	79.66	96.8	98.49
20	83.10	85.38	98.8	98.91





CONCLUSIONS

- As is evident from the results, user location is an important input feature for the model, which on inclusion, produces more than double of top-1 and top-3 accuracies given by the author's architecture for the lowest SNR.
- Using EarlyStop helped prevent overfitting and saved training time & computation for the -10dB and 0dB SNR cases because the model stopped training before it reached the 100 epoch mark.
- For SNRs more than 0dB, the model has classified almost all the classes perfectly in the top-3 beams. That means, instead of beam training, the device may simply connect to those three beams one by one and see which one gives the best rate.

REFERENCES

1. “Deep Learning for mmWave Beam and Blockage Prediction Using Sub-6GHz Channels” by Muhammad Alrabeiah and Ahmed Alkhateeb
2. “DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications” by Ahmed Alkhateeb
3. www.deepMIMO.net
4. https://github.com/malrabeiah/Sub6-Preds-mmWave
5. “A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies” by Marco Giordani *et al*
6. “Beam Management in Millimeter-Wave Communications for 5G and Beyond” by Yu-Ngok Ruyue Li *et al*