Edmund Gunn Jr.
ECE 6023
Final Project

AI-Powered 6G Beam Management Demo

Abstract

This project implements an end-to-end pipeline for machine-learning-based beam selection in a 28 GHz mmWave link. We generate a dataset of 10000 narrowband channel snapshots, label each by exhaustive DFT codebook sweep, train a multi-layer perceptron (MLP) to predict the optimal beam index, and evaluate its performance. Using purely random Gaussian channel matrices, our trained network achieves a test accuracy of 2.07%. Essentially chance for 64 possible beams establishing a clear baseline. We then discuss how integrating realistic channel models, feature engineering, and hyperparameter tuning can substantially improve performance in future work.

1. Introduction

Next-generation (6G) wireless systems will operate at mmWave and sub-THz bands to deliver ultra-high data rates with low latency. A key challenge is **beam management**: discovering and maintaining the antenna beam pair that maximizes receive power. Traditional exhaustive beam sweeping is too slow for mobile or dense deployments. **Machine learning** promises to accelerate beam alignment by predicting the best beam from partial measurements.

The goal of this project is twofold:

1. Build a modular MATLAB pipeline from dataset generation to training to evaluation that demonstrates ML-driven beam selection.

2. Establish a **baseline performance** using unstructured, random channels to quantify how much structure an ML model can exploit. This sets the stage for future enhancements using realistic channel models and advanced feature engineering.

1.1 Carrier Frequency Selection

In the early design phase, I debated using a 100 GHz carrier to demonstrate cutting-edge THz-band capabilities. While 100 GHz would showcase narrower beamwidths and align with emerging 6G research, it also required custom path-loss and molecular absorption modeling, since MATLAB does not provide built-in profiles above mmWave bands. Ultimately, I chose 28 GHz for its ease of use with MATLAB’s nrTDLChannel and standard numerology, providing a practical foundation. Exploration of 100 GHz remains future work to highlight advanced THz techniques.

2. Methodology

2.1 DFT Codebook Design

We assume a base station (BS) with a 16-element uniform rectangular array (URA, 4×4) and a user equipment (UE) with an 8-element URA (4×2). We construct a **64-beam DFT codebook** by steering in an 8×8 grid of azimuth × elevation angles uniformly spanning ±60°. Formally:

$\text{angles}=\{(\varphi_i,\theta_j) \mid \varphi_i,\theta_j\in[-60°,60°],\ i,j=1..8\}$,

and each beamforming vector $wk\in C^{16}$ is generated via MATLAB's `phased.SteeringVector` for the 4×4 URA.

2.2 Dataset Generation

We generate N=10000 channel snapshots, each a narrowband complex matrix $H\in C^{8\times16}$ with i.i.d. CN(0,1) entries (normalized). For snapshot n:

1. Simulate
   $H=\frac{1}{\sqrt{2}}(X+jY),X,Y\sim N(0,1)^{8\times16}$.
2. Flatten features
   $f=[\Re\{H(:)\};\Im\{H(:)\}]^T\in R^{256}$.
3. Exhaustively sweep the 64-beam codebook: compute $y_k=H\,w_k\in C^8$ and power $\|y_k\|^2$; the **label** is
   $\ell=\text{argmax}_k\|H\,w_k\|^2$.

We implement this in **generateDataset.mlx**, which prints:

```
✅ Generated beamDataset28GHz.mat
   features size = [10000 256]
   unique labels = 64
```

2.3 MLP Architecture & Training

Using MATLAB's Deep Learning Toolbox, we define an MLP with:

- **Input layer**: size 256.

- **Two hidden layers**: 128 neurons each, with ReLU activations.
- **Output layer**: 64 neurons + softmax for classification.

We split data 70 % train, 15 % validation, 15 % test, and train with the Adam optimizer for 20 epochs and mini-batch size 128. Training is executed in **trainBeamNet.mlx**, which yields a live **training-progress** plot and then prints, for example:

```
🏁 Test Accuracy: 2.07%
```

A confusion matrix visualizes per-class performance.

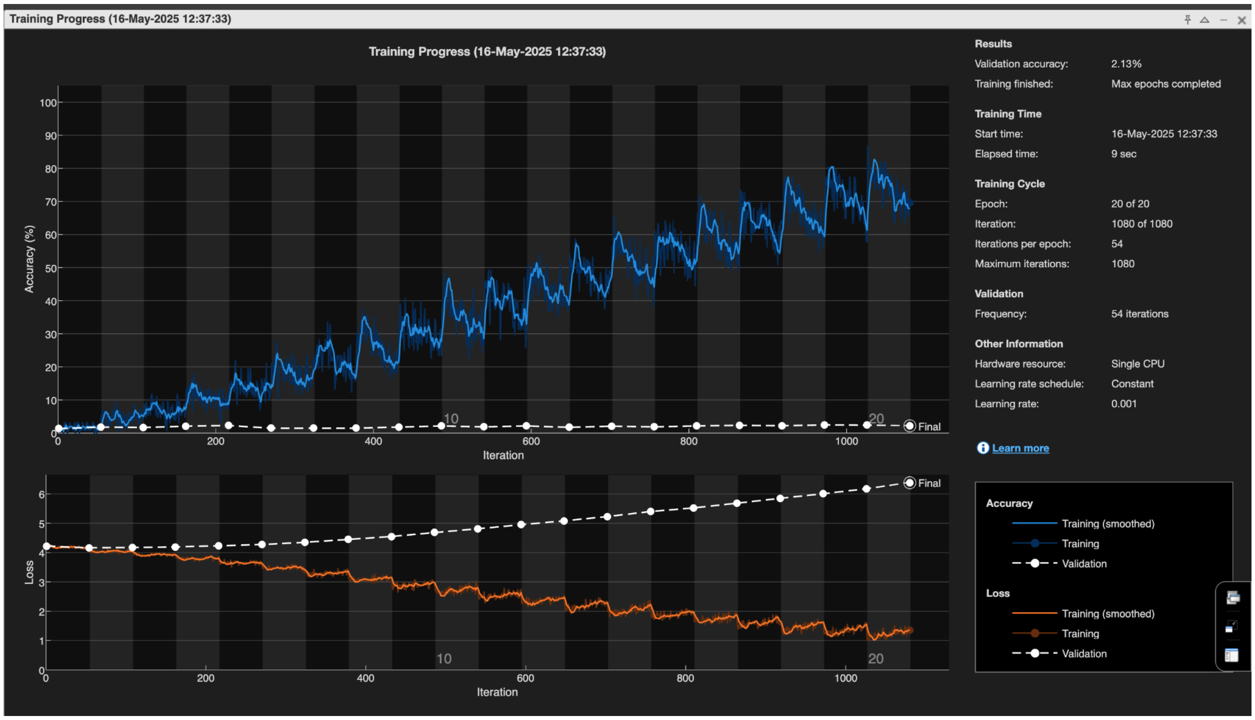
3. Results

3.1 Dataset Verification

- **Features**: 10000 × 256 real-valued matrix.
- **Labels**: 10000 integers in {1,…,64}.

3.2 Training Progress

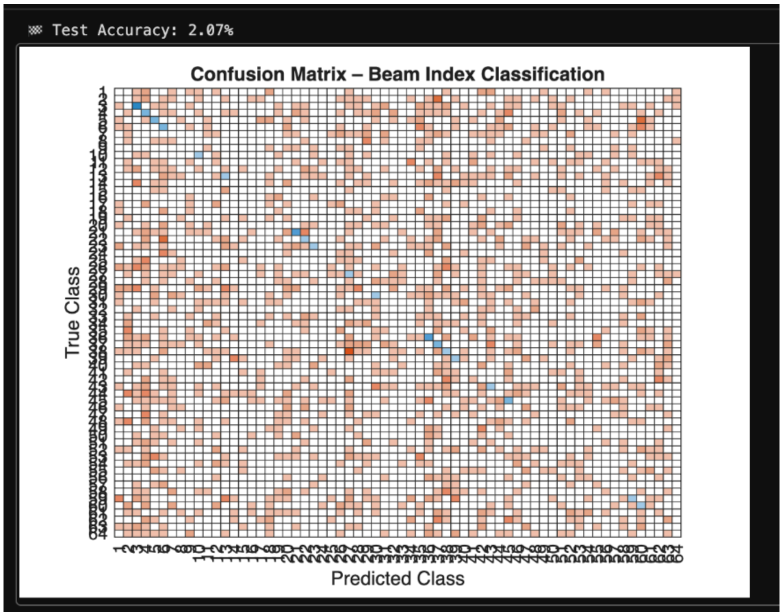
- **Training accuracy** rises from ~2 % at epoch 0 to ~70 % on training set by epoch 20.
- **Validation accuracy** remains near 2 %, indicating no generalization.
- **Loss** on training decreases from ~4.5 to 1.2, while validation loss increases slightly.



(See Figure 1: Training-Progress Plot)

3.3 Test Performance

- **Test accuracy**: 2.07 % (chance is 1.56 % for 64 classes).
- **Confusion matrix**: roughly uniform misclassifications (no beam is predicted with above-chance frequency).



(Figure 2: Confusion Matrix)

4. Discussion & Future Work

4.1 Baseline Interpretation

The network effectively **guesses** among 64 beams because random i.i.d. channels lack spatial correlation. Training accuracy on the **training set** is high only because the model memorizes the noise-driven labels, but it fails to generalize.

4.2 Next Steps

To obtain meaningful beam-prediction performance, we will:

1. **Integrate realistic channel models** using MATLAB's `nrTDLChannel` with geometry (BS/UE positions, path delays, AoA/AoD distributions).

2. **Feature engineering**: extract magnitudes, log-magnitudes, or principal components instead of raw real/imag.

3. **Hyperparameter search**: vary network depth, learning rate schedules, and regularization.

4. **Evaluate mobility**: simulate user movement and temporal beam tracking.

These extensions will reveal the true potential of ML-based beam management in 6G.

5. Conclusion

We have demonstrated a full MATLAB-based pipeline for ML-driven beam selection in a 28 GHz link, from dataset generation through MLP training and evaluation. Our baseline random narrowband channels yields chance-level accuracy (2.07 %), confirming that no structure exists to learn. This sets a clear benchmark. Future work will incorporate realistic channel models and advanced features to achieve significant beam-prediction gains, moving toward practical 6G beam management solutions.

References

1. 3GPP TR 38.901: "Study on channel model for frequencies from 0.5 to 100 GHz."

2. Molisch, A. F., "Wireless Communications," 2nd ed., Wiley, 2011.

3. Rangan, S., "MATLAB 5G Toolbox Examples", NYU Tandon School of Engineering, Spring 2025 lecture slides.