Deep Learning Based High Accuracy Real World Localization in Indoor mmWave Networks With Virtual Anchors

Seyed Alireza Javid*, Soroush Mesforush Mashhad*, Nima Razavi*
School of Electrical and Computer Engineering
University of California, San Diego
San Diego, CA, USA
{sjavid, smesforushmashhad, nrazavi}@ucsd.edu

***Abstract*—Milimeter-wave (mmWave) technology has revolutionized the realm of localization. The sparsity of the channel enables us to use channel estimation methods to estimate the parameters needed for localization. We try to provide a deep-learning based algorithm for high-accuracy indoor localization using the mirror images of a single access point (AP) generated by specular reflections from the room environment which will be walls, ceilings, and floor. These mirror images are called virtual anchors (VA). For data collection, we attempted to use the mmWave testbed M-cube. We have attempted to utilize an extended Gaussian mixture model (GMM) method and in the end, a one-shot deep learning algorithm to refine user position estimates.**

***Index Terms*—Localization, mmWave, Virtual Anchor(VA), Deep Learning(DL), Gaussian Mixture Model(GMM)**

I. Introduction

A. Motivation

It is widely known that the emergence of mmWave technology has revolutionized the realms of communication and sensing. Localization, which is in this spectrum, has also been impacted by this breakthrough. In localization, we require parameters related to the channel paths such as Angle of Arrival (AoA), Angle of Departure (AoD), and the delays. Since we are working at mmWave band, we can leverage sparse channel estimates providing information about the path parameters [1].

Accurate indoor localization is an important factor for many applications. For example, in an indoor factory scenario, where there is a lot of commotion, accurate positioning is essential for navigation. Other locations such as airports, malls, and hospitals also have the same concerns when it comes to position estimation. Localization also improves resource allocation for networks by enabling location-aware services. Many existing methods incorporating mmnWave methods are based on highly idealized conditions that do not capture the true nature of a real world scenario. An important motivation for our work is bridging the gap between these works and real world scenarios.

*These authors contributed equally to this work.

B. Problem Statement

The efforts in this work are focused on an indoor localization challenge with a single access point (AP). The primary goal is to estimate user position by learning the locations of the common virtual anchors (VAs) which are mostly static in indoor environments, by using the channel data collected at different user locations. Achieving this requires the reliable extraction of key channel parameters from the mmWave signal, a task complicated by factors such as multipath propagation, interference, and the inherent variability of indoor environments. Our challenge is to design a system that can leverage the sparse nature of mmWave channels while accounting for real world imperfections. The solution must robustly distinguish between useful virtual anchors and spurious reflections We will utilize machine learning (ML) and deep learning (DL) techniques for this stage. We will also leverage the learned virtual anchor locations for precise localization of the users. We design a two-stage indoor localization algorithm to estimate user positions based on the AoA and relative delays of the channel paths at the AP. These AoAs and relative delays are assumed to be obtained using a sparse recovery algorithm as described in [2].

C. Prior Work

Previous works have given promising results of sparse recovery for mmWave localization by extracting the needed parameters for localization via algorithms such as multi-dimensional ESPRIT [1], [2]. Many of these approaches are evaluated under simple conditions. In the real world scenario, we are dealing with problems such as non-line of sight (NLoS), dynamic interference, clutter, and unpredictable multipath components. Also, evaluating static virtual anchor locations has remained a significant challenge in previous literature. Many methods cannot distinguish between real reflection points and those generated by scattering, clutter, or other environmental deficiencies. Hence, there is a gap between this promising technology and real world systems.

D. Key Ideas

To overcome limitations discussed, our work is based on a two-stage localization algorithm combined with deep learning methods. Our contributions are summarized as follows.

- First, data will be generated using a real-world mmWave testbed, M-Cube [3]. Previous works with mmWave indoor localization using virtual anchor points do not evaluate their methods on real-world implementation. Our goal is to generate data using the M-Cube testbed to apply further enhancements to existing virtual anchor-based localization.
- After collecting the data, the initial localization algorithm capitalizes on the observation that first-order reflections predominantly originate from vertical walls or horizontal surfaces, such as floors and ceilings, in typical indoor environments.
- Next, the most frequently occurring static virtual anchor locations are identified by analyzing the initial location estimates of users across various positions. This step involves leveraging deep learning techniques applied to real-world data for improved accuracy and robustness.
- Finally, we propose a one-shot localization algorithm based on deep learning methods that utilize path parameters and virtual anchor locations to enhance the localization process efficiently and we evaluate our method against [4].

II. Related Work

A. Single AP Localization

Single AP user localization is a hot research topic for mmWave [5]. However, most of the literature makes hard and unrealistic assumptions like perfect time synchronization between the transmitter and receiver [6]. This does not happen in any practical system. In practice, an unknown clock offset (CO) should be considered to make the localization procedure more realistic [2]. There are many methods and algorithms to compensate for CO. One way, is to solve a system of equations that leverages the geometric relationship between the user location and path parameters. In [2], the properties of the indoor environment are exploited to estimate the CO and then proceed with localization. However, erroneous AoA, AoD, and relative delay estimates degrade the performance of the algorithms in [2]. In [7], a stochastic localization algorithm is proposed using the fact that the virtual anchor location created by a reflector remains constant for different locations of the user, provided that the same reflector creates a path. We aim to use a similar approach in this project for indoor localization.

B. Hardware Based Localization

Many papers are trying to do indoor localization via virtual anchors using hardware. The authors in [8] focus on multi-bounce scattering to sense objects. Their approach while being innovative, has a complex multi-layer pipeline and is heavily dependent on unrealistic ideal models. To elaborate, on the multi-bounce method, they must consider first, second, and third-order reflections with matched filtering. Whereas, we use a straightforward geometric model where we use AoAs, delays, and specular reflection models to compute virtual anchor locations. In [8], sequential processing of multiple bounce orders is essential which is highly complex. We completely sidestep the sequential processing and by relying on the sparse channel property, we extract the channel parameters and by feeding it into the neural network we aim to directly localize the users. In Hawkeye [9], they perform localization in the hectometer range. The performance of this work under multi-path and clutter is not applicable for practical use cases, and the efficacy highly depends on a particularly set frequency. We believe that by using deep learning, we can increase robustness compared to this and similar works. In SuperSight [10], indoor localization is attempted with a triangular tag array and a concept called virtual radar formation. Extracting such virtual radar requires AoA and measured range information, and the formula used is delicate, making it very sensitive to even minor errors in the measurements, whereas our approach using virtual anchors will be more sturdy. The authors in [11] develop a system to enhance mmWave localization using hardware. Since mmWave is being applied, the channels are sparse, meaning that distinguishing between line of sight (LoS) and powerful non-line of sight (NLoS) is particularly difficult. Due to the binary decision making in the system, this fact makes it prone to error, so in some sense, channel sparsity which is normally leveraged in the literature about mmWave, is a disadvantage in this paper. Also, the hardware used and the modifications made are not extendable to commercial products.

C. Neural Network Based Localization

Some literature can be found implementing neural networks to aid the localization process. The authors in [12] implement a neural network-based algorithm to refine localization estimates and test in a real-world setting using the virtual anchoring method. In contrast to their paper, our implementation of mmWave localization with virtual sources does not assume synchronized clocks which is a much more practical assumption. Our deep learning architecture is used for CO estimation as well, a much harder problem in real-world settings. Additionally, it does not incorporate the COMPAS inference engine introduced in [7], a situationally aware mmWave MIMO engine aiding localization. The authors in [13] use known 3-D maps of the indoor area to predict and create the optimal virtual anchor points and then perform localization. In contrast, our work does not attempt any optimization of anchor point placement. We shall utilize the position of the anchor points given.

III. Design

After evaluating the initial location of the users, we try to estimate virtual anchors relying on the specular geometry. The virtual image of the AP, reflected by the surface corresponding to the l-th path can be expressed as $\mathbf{a}'_l =$

$\mathbf{u} + c\left(\Delta\tau_l + \tau_0\right)\left(-\boldsymbol{\theta}_l + 2\mathbf{n}_l \left\langle \mathbf{n}_l, \boldsymbol{\theta}_l \right\rangle\right)$, where $\mathbf{n}_l = \frac{\mathbf{u}-\mathbf{u}'_l}{\left\|\mathbf{u}-\mathbf{u}'_l\right\|}$ is the normal vector. By repeating this process for different user locations, we can construct a set of virtual anchor positions. In indoor environments, strong first-order reflections arise from walls, floors, ceilings, and static objects, making it crucial to identify the most frequently occurring virtual anchor locations. However, the estimated locations may include redundant reflections, spurious estimates caused by scattering rather than specular reflection, and occasional errors due to localization outages. We must select only a set of dominant virtual anchors to ensure robustness. We utilize Gaussian Mixture Models (GMM) for anchor locations. Consider the following equation.

$$f(\mathbf{a}_l) = \sum_{i=1}^{M} w_i g\left(\mathbf{a}_l \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \tag{1}$$

where $\mathbf{a}_l \in \mathbb{R}^3$, $w_i \in \mathbb{R}$ are the mixture weights and $\sum_{i=1}^{M} w_i = 1$. The 3-dimensional mixture densities are defined as follows.

$$g\left(\mathbf{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) = \frac{1}{(2\pi)^{D/2}\left|\Sigma_i\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_i\right)^T \Sigma_i^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_i\right)\right\} \tag{2}$$

with mean vector $\boldsymbol{\mu}_i \in \mathbb{R}^3$ and covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times 3}$. These values can be calculated using the Expectation Maximization (EM) algorithm. Using this method, $\boldsymbol{\mu}_i$s will describe the estimated locations of anchors. Here, M is a hyperparameter that indicates the number of anchors in the problem. However, the points do not have an equal contribution to selecting the virtual anchor. We define a weight for each point based on their power. The overall algorithm is as depicted in 1.

To simplify the process, once the virtual anchor locations are computed, we define the anchor set as the collection of selected virtual anchors along with the AP.

Algorithm 1 Iterative Regularized Gaussian Mixture Model (GMM)

Input: Virtual anchor positions $\mathbf{X} = \{x_i\}_{i=1}^N$, weights $\mathbf{w} = \{w_i\}_{i=1}^N$, maximum iterations $T_{\max}$, convergence threshold ϵ.
Initialize: Set iteration counter $t = 0$, define number of clusters K, initialize Gaussian Mixture Model (GMM) with K components.
Fit GMM: Train GMM on dataset $\mathbf{X}$ using Expectation-Maximization (EM).
repeat
  Assign each point x_i to its cluster:

$$z_i = \arg\max_k P(k \mid x_i)$$

  Compute the weighted mean for each cluster:

$$\mu'_k = \frac{\sum_{i:z_i=k} w_i x_i}{\sum_{i:z_i=k} w_i}$$

  Compute the difference between new and old means:

$$d_k = \|\mu'_k - \mu_k\|$$

  if $\max_k d_k < \epsilon$ **then**
    Converged, exit loop.
  end if
  Update GMM cluster means:

$$\mu_k = \mu'_k$$

  Refit GMM using updated means.
  Increment iteration counter: $t \leftarrow t + 1$.
until $t = T_{\max}$ or convergence is reached
Output: Converged cluster centers $\{\mu_k\}_{k=1}^K$ as virtual anchor locations.

A. Learning-Based Localization with Virtual Anchors

Let the ℓ-th path correspond to a reflection associated with the anchor $\mathbf{a}'_a \in \mathcal{A}$, where $\mathcal{A}$ denotes the set of selected anchors. The user location can be expressed as:

$$\mathbf{u} = \mathbf{a}'_a + c\left(\Delta\tau_\ell + \tau_0\right)\boldsymbol{\theta}'_\ell, \tag{3}$$

Where the specular reflection AoA is given by:

$$\boldsymbol{\theta}'_\ell = \boldsymbol{\theta}_\ell - 2\mathbf{n}_a \left\langle \mathbf{n}_a, \boldsymbol{\theta}_\ell \right\rangle. \tag{4}$$

Here, $\mathbf{n}_a$ is the normal vector to the surface that induces the specular reflection and is defined as:

$$\mathbf{n}_a = \frac{\mathbf{a}'_a - \mathbf{a}}{\|\mathbf{a}'_a - \mathbf{a}\|}. \tag{5}$$

Note that the AP location is included in the set of anchors by defining $\boldsymbol{\theta}'_\ell = \boldsymbol{\theta}_\ell$ for the LOS scenario. We show the set of pairs $P = \left\{(l,a) \in \mathbb{N}^2 \text{ such that path } l \text{ comes from the anchor } \mathbf{a}'_a\right\}$.

1) Problem Formulation: Finally, the optimization problem using a minimum mean square error (MSE) solution is given as

$$\min_{\mathbf{u},\tau_0,P} \sum_{(l,a)\in\mathcal{P}} \left\|\mathbf{u} - \left(\mathbf{a}'_a + \boldsymbol{\theta}'_l c\left(\Delta\tau_l + \tau_0\right)\right)\right\|^2$$

Before introducing the solution we evaluate $\hat{\mathbf{u}}$ for a given P and $\hat{\tau}_0$ as follows

$$\hat{\mathbf{u}} = \frac{1}{|\mathcal{P}|}\sum_{(l,a)\in\mathcal{P}} \mathbf{a}'_a + c\left(\Delta\tau_l + \tau_0\right)\boldsymbol{\theta}'_l$$

Using this value for a given P, we evaluate delay as

$$\hat{\tau}_0 = \frac{\sum_{(l,a)\in\mathcal{P}}\left(c\Delta\tau_l + \left\langle \mathbf{a}'_a, \boldsymbol{\theta}'_l\right\rangle\right)}{|\mathcal{P}|c - \frac{1}{|\mathcal{P}|}\sum_{(l,a),(l',a')\in\mathcal{P}} c\left\langle \boldsymbol{\theta}'_{l'}, \boldsymbol{\theta}'_l\right\rangle} + \frac{\frac{1}{|\mathcal{P}|}\sum_{(l,a),(l',a')\in\mathcal{P}}\left(c\Delta\tau_l\left\langle \boldsymbol{\theta}'_{l'}, \boldsymbol{\theta}'_l\right\rangle + \left\langle \mathbf{a}'_{a'}, \boldsymbol{\theta}'_l\right\rangle\right)}{|\mathcal{P}|c - \frac{1}{|\mathcal{P}|}\sum_{(l,a),(l',a')\in\mathcal{P}} c\left\langle \boldsymbol{\theta}'_{l'}, \boldsymbol{\theta}'_l\right\rangle}.$$

2) Deep Learning Method: Although we have formulated the optimized values for the location and delay directly within the optimization problem, our formulation assumes a given set

of anchor-path pairs, denoted as P. However, to accurately localize the users, we must first determine this set P.

To achieve this, we employ a supervised feedforward neural network to learn and predict the anchor-path pair assignments, enabling an effective localization framework. The following table demonstrates the characteristics of this network. The

TABLE I
NEURAL NETWORK ARCHITECTURE FOR MMWAVE INDOOR LOCALIZATION

Layer	Type	Output Shape	Parameters
Input	Input Layer	(None, 25)	0
1	Dense	(None, 128)	3,328
2	Batch Normalization	(None, 128)	512
3	Dropout (0.3)	(None, 128)	0
4	Dense	(None, 64)	8,256
5	Batch Normalization	(None, 64)	256
6	Dropout (0.2)	(None, 64)	0
7	Dense	(None, 32)	2,080
8	Dense	(None, 3)	99
Total Trainable Parameters			14,531

MSE loss function for this network can be defined as

$$\sum_{(l,a)\in\mathcal{P}} \left\| \mathbf{u} - \left(\mathbf{a}'_a + \boldsymbol{\theta}'_l c\left(\Delta\tau_l + \tau_0\right)\right)\right\|^2$$

To apply this method, we must first label the anchors, classify the corresponding paths, and determine the set of anchor-path pairs P.

In a nutshell, the algorithm used is as follows.

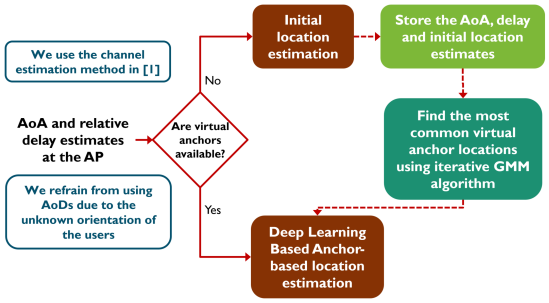

Fig. 1. Proposed algorithm

IV. HARDWARE

The experimental testbed for this project is built around the M-Cube mmWave software-defined radio (SDR) platform [3], a mmWave testbed for communication experiments, as seen in Figure 2. The M-Cube platform utilizes a router with phased antenna arrays, real-time beamforming capabilities, and down-conversion modules that input to a conventional sub 6 GHz SDR. Since conventional SDRs cannot directly process mmWave frequencies (typically above 6 GHz), the M-Cube includes a dedicated bridge converter that down-converts the mmWave signals to an intermediate frequency (sub-6 GHz). This conversion preserves the essential characteristics of the signal while enabling subsequent processing using standard baseband hardware. Additionally, the testbed ensures the sparsity inherent in mmWave channels, a feature that is central to our localization strategy via virtual anchors. Although the M-Cube testbed incorporates a phased array capable of electronically steering the beam in real time, we kept the beams fixed across all measurements. The firmware that controls the array is in 12 and the values of the phases and magnitudes are confirmed in Figure 4. The down-converted baseband signals are fed into a USRP N210 SDR [14] (seen in Figure 5), which is the recommended software radio for interfacing with the baseband module, and we output its connection and details in Fig. 6. In our testbed, the USRP performs upconversion (at the transmitter) and downconversion (at the receiver), in conjunction with GNU Radio for real-time control and data processing. GNU Radio is used as the primary framework for both data acquisition and real-time processing. The flexibility of GNU Radio allows for seamless integration of the digital baseband processing. The end-to-end signal chain in the M-Cube testbed is designed to capture all the essential parameters for localization. First, a random binary sequence is generated and modulated using QPSK. This sequence, with known properties (including a fixed random seed known to the receiver), serves as the pilot signal for channel estimation. MATLAB is used to generate this baseband signal, which is then saved in a format suitable for further processing. The generated baseband signal is loaded into GNU Radio, where it is interfaced with the USRP N210. The USRP performs digital-to-analog conversion and upconverts the signal to the designated RF carrier. Although the M-Cube platform provides additional beamforming and phasing capabilities, the final upconversion to RF is handled by the USRP. The upconverted RF signal is transmitted over the air and then captured by the M-Cube's mmWave front-end. The bridge converter down-converts the received mmWave signals to an intermediate frequency, effectively translating the high-frequency information into a digital-friendly band without significant loss of fidelity. At the receiver, the USRP N210 captures the down-converted signal, performs further downconversion to the baseband, and digitizes the signal. GNU Radio then saves the digitized baseband data (which contains the channel response and hardware impairments) for offline processing and channel estimation. In collecting training data, we note the transmitter's location relative to the receiver. This spatial information is crucial for mapping virtual anchors, as it helps distinguish between the direct path and reflected paths.

A. Experimental Setup and Environmental Considerations

Experiments were conducted in a controlled indoor environment, specifically a conference room that is representative of typical indoor settings. The transmitting SDR, integrated with the M-Cube testbed, was moved to multiple locations within the room to emulate various user positions. This mobility is critical to capturing the spatial diversity of the channel and to reliably map the locations of virtual anchors. Experiments indicated that operating at 1 GHz resulted in significant interference from other transmitters in the vicinity. To mitigate

this issue, the system was configured to operate at 902 MHz, which frequency that demonstrated minimal interference and provided a cleaner channel for accurate parameter estimation. As seen in Figure 10, no other user is operating at that frequency. The transmitter's spectrogram is shown in Figure 9 and a sample received spectrogram is shown in Figure 11. A sampling rate of 1 MHz was chosen to ensure adequate resolution for delay estimation without overwhelming the processing pipeline. Each antenna element was configured to operate at maximum gain to enhance the signal-to-noise ratio (SNR), critical for resolving weak multipath components. A firmware controller (Fig. 3) maintains tight synchronization between the transmitting and receiving units via a dedicated network router and ensures the samples from each antenna array module are synchronized. A protocol was created to calculate the necessary channel parameters and training data points. A transmitter first generates the baseband waveform utilizing known pilot QPSK symbols. The transmitted signal data is saved in a file and read by GNU Radio. The receiver file performs time synchronization, and CFO correction, and estimates the angle of arrival, degrees of freedom, and power of the delayed paths. Time and CFO synchronization utilize known pilot symbols and assume the CFO drift is minimal across each packet. The starting index of the first packet where the cross-correlation is maximized. The starting index of the reflected path determines the path delays and the power of the delayed signal using correlation with the known pilot sequence. The receiver collects the input variables needed for the next process.

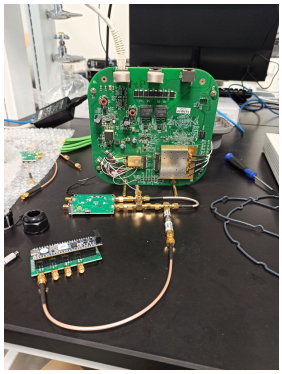

Fig. 2. The M-Cube testbed showcasing the integrated mmWave front-end, bridge converter, and phased antenna array.

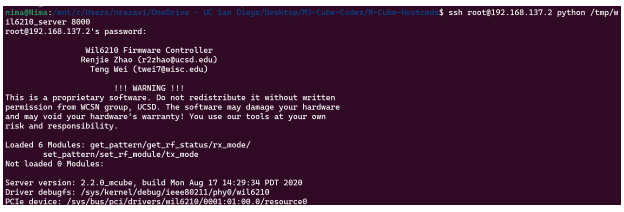

Fig. 3. Firmware Controller interface that manages antenna array synchronization and facilitates control over the testbed via a network router.

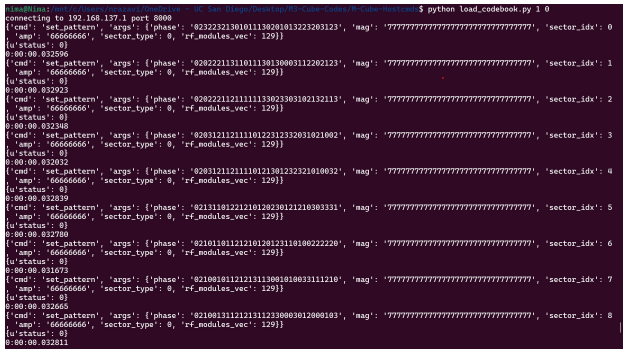

Fig. 4. Beamforming codebook ensuring that magnitudes and phases across all array modules are the same

Fig. 5. Ettus research USRP N210 SDR

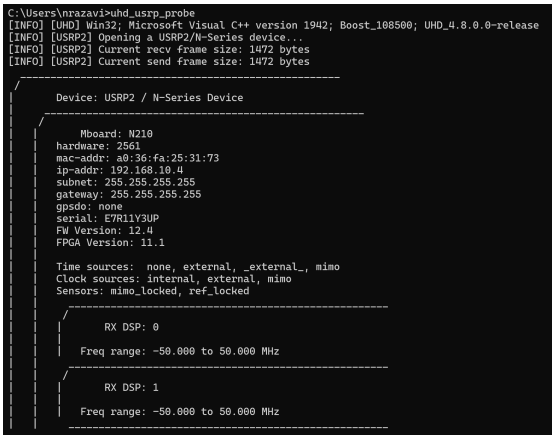

Fig. 6. Connection test from laptop to USRP

V. Evaluation and Results

After gathering data, proceeded to estimate the positions of the virtual anchors via the algorithm presented in the previous

sections, one of the setups we have gathered data from is depicted in Fig. 7.

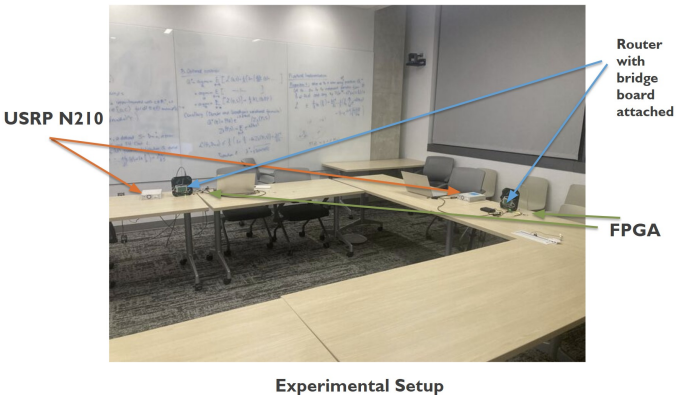

Fig. 7. Experimental setting for data collection

After virtual anchor estimation, we proceeded to carefully implement the theoretical method we have explained. We have compared our results against [4] as a benchmark and also against the initial anchor version. We have deployed our neural network architecture with 100 epochs and a learning rate of $10^{-}3$, with and without virtual anchors. We expect the presence of the virtual anchors to enhance the localization performance compared to all other methods. The localization empirical error CDFs are obtained after training the respective neural networks and also implementing [4] and the initial algorithm is depicted in 8.

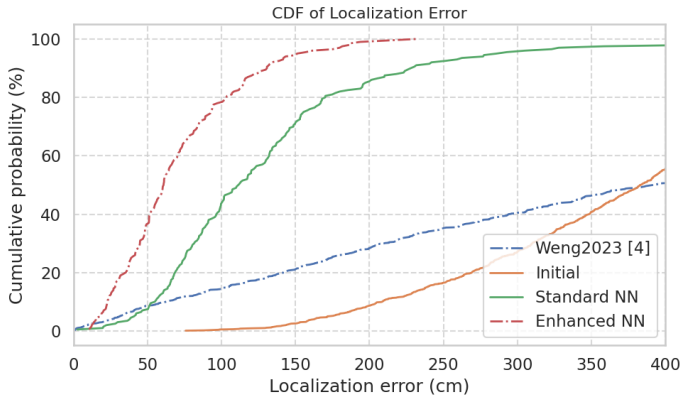

Fig. 8. Localization empirical error CDF

It is evident that our approach marginally improves upon the existing approaches. The integration of the virtual anchors that are extracted through a fine-tuned GMM method with the deep learning-based approach delivers promising and improved results. We see a substantial improvement compared to the benchmark and also compared to the scenario where virtual anchors are not leveraged. We see that our method can perform with lower localization errors which is convenient for indoor localization situations. It is also important to mention that the incorporation of deep learning by nature reduces the theoretical difficulty of the problems and allows us to fine-tune in many situations with try and error which is practical for real-world scenarios and we do not need the data to fit a predefined, generalized and sometimes unrealistic statistical model.

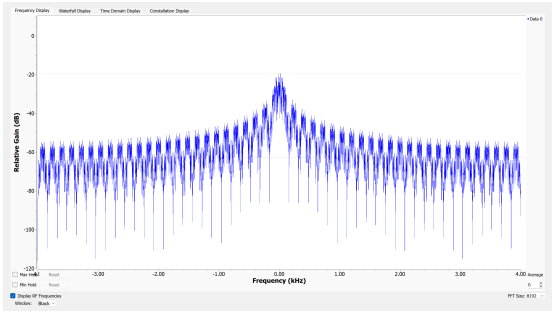

Fig. 9. The spectrogram at one point in time of the transmitted signal

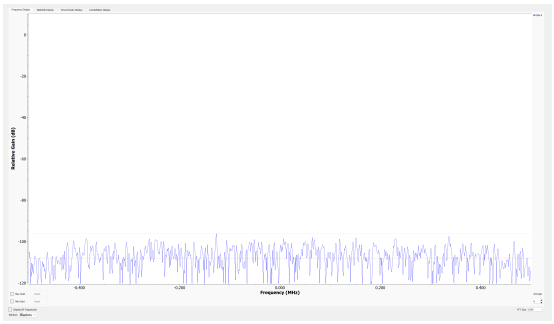

Fig. 10. The received spectrogram when the transmitter is not on

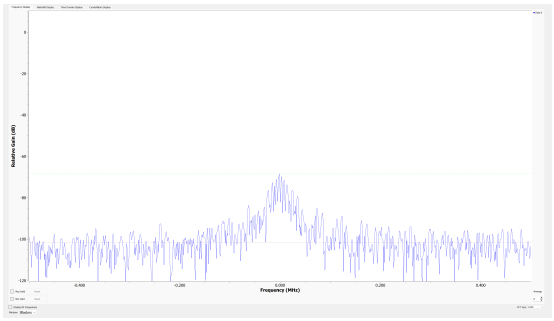

Fig. 11. The received spectrogram when the transmitter is on

Fig. 12. The Firmware Controller that establishes server connection to the router to control the antenna array

VI. Conclusion

In conclusion, in this literature, we have introduced an indoor mmWave localization by leveraging virtual anchors and deep learning. We first identify virtual anchor locations using an iterative GMM and then integrate the said virtual anchors into a carefully designed deep-learning framework that refines user position estimates. We have gathered real-world data using the mmWave testbed M-Cube, by performing channel estimation on the gathered data, we proceeded to localization where the localization errors were reduced compared to baseline methods that rely only on line-of-sight or single-reflection assumptions. Furthermore, our results highlight the effectiveness of applying data-driven techniques to real-world channel measurements, even when confronted with imperfect synchronization, varying interference, and the multipath complexity typical of indoor environments. For future works, one may extend this research to multi-AP networks, considering higher orders of reflection, and more sophisticated neural network models to enhance localization accuracy.

VII. Team Contribution

This project has been a team collaboration, We have done our utmost best to distribute tasks equally and based on expertise. To be precise, Soroush's impact has been on the localization techniques and formulation of the problem. Alireza's expertise and insight in deep-learning have been pivotal in designing the neural network model required in this project. Nima's proficiency in hardware and previous experience in real-world setup has been crucial in this project. Regarding other tasks such as literature survey, debugging problems, and writing the midterm and final reports, all members of the team have contributed equally and to the best of their knowledge and expertise.

VIII. Acknowledgments

We would like to thank ChatGPT, Claude.ai, and Grammarly. These tools greatly helped us refining various aspects of our report.

References

[1] A. Shahmansoori, G. E. Garcia, G. Destino, G. Seco-Granados, and H. Wymeersch, "Position and orientation estimation through millimeter-wave mimo in 5g systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1822–1835, 2017.

[2] J. Palacios and N. González-Prelcic, "Separable multidimensional orthogonal matching pursuit and its application to joint localization and communication at mmwave," in *2023 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2023, pp. 1421–1426.

[3] R. Zhao, T. Woodford, T. Wei, K. Qian, and X. Zhang, "M-cube: a millimeter-wave massive mimo software radio," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3372224.3380892

[4] S. Weng, F. Jiang, and H. Wymeersch, "Wideband mmwave massive mimo channel estimation and localization," *IEEE Wireless Communications Letters*, vol. 12, no. 8, pp. 1314–1318, 2023.

[5] J. Palacios, G. Bielsa, P. Casari, and J. Widmer, "Single-and multiple-access point indoor localization for millimeter-wave networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1927–1942, 2019.

[6] J. Li, M. F. Da Costa, and U. Mitra, "Joint localization and orientation estimation in millimeter-wave mimo ofdm systems via atomic norm minimization," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4252–4264, 2022.

[7] R. Mendrzik, F. Meyer, G. Bauch, and M. Z. Win, "Enabling situational awareness in millimeter wave massive mimo systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 1196–1211, 2019.

[8] N. Mehrotra, D. Pandey, A. Prabhakara, Y. Liu, S. Kumar, and A. Sabharwal, "Hydra: Exploiting multi-bounce scattering for beyond-field-of-view mmwave radar," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 1545–1559.

[9] K. M. Bae, H. Moon, S.-M. Sohn, and S. M. Kim, "Hawkeye: Hectometer-range subcentimeter localization for large-scale mmwave backscatter," in *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, 2023, pp. 303–316.

[10] K. M. Bae, H. Moon, and S. M. Kim, "Supersight: Sub-cm nlos localization for mmwave backscatter," in *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, 2024, pp. 278–291.

[11] A. Blanco, P. J. Mateo, F. Gringoli, and J. Widmer, "Augmenting mmwave localization accuracy through sub-6 ghz on off-the-shelf devices," in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, 2022, pp. 477–490.

[12] A. Shastri, S. Blandino, C. Gentile, C. Lai, and P. Casari, "Algorithm-supervised millimeter wave indoor localization using tiny neural networks," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2025.

[13] O. Kanhere, S. Ju, Y. Xing, and T. S. Rappaport, "Map-assisted millimeter wave localization for accurate position location," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[14] a. N. I. B. Ettus Research, "USRP N310 — ettus.com," https://www.ettus.com/all-products/usrp-n310/, [Accessed 13-02-2025].