

# A STRUCTURED DEEP NEURAL NETWORK FOR DATA-DRIVEN LOCALIZATION IN HIGH FREQUENCY WIRELESS NETWORKS

Marcus Z. Comiter<sup>1</sup>, Michael B. Crouse<sup>1</sup> and H. T. Kung<sup>1</sup>

<sup>1</sup>John A. Paulson School of Engineering and Applied Science  
Harvard University

Cambridge, MA 02138

[marcuscomiter@g.harvard.edu](mailto:marcuscomiter@g.harvard.edu)

[mcrouse@g.harvard.edu](mailto:mcrouse@g.harvard.edu)

[kung@harvard.edu](mailto:kung@harvard.edu)

## ABSTRACT

*Next-generation wireless networks such as 5G and 802.11ad networks will use millimeter waves operating at 28GHz, 38GHz, or higher frequencies to deliver unprecedentedly high data rates, e.g., 10 gigabits per second. However, millimeter waves must be used directionally with narrow beams in order to overcome the large attenuation due to their higher frequency. To achieve high data rates in a mobile setting, communicating nodes need to align their beams dynamically, quickly, and in high resolution. We propose a data-driven, deep neural network (DNN) approach to provide robust localization for beam alignment, using a lower frequency spectrum (e.g., 2.4 GHz). The proposed DNN-based localization methods use the angle of arrival derived from phase differences in the signal received at multiple antenna arrays to infer the location of a mobile node. Our methods differ from others that use DNNs as a black box in that the structure of our neural network model is tailored to address difficulties associated with the domain, such as collinearity of the mobile node with antenna arrays, fading and multipath. We show that training our models requires a small number of sample locations, such as 30 or fewer, making the proposed methods practical. Our specific contributions are: (1) a structured DNN approach where the neural network topology reflects the placement of antenna arrays, (2) a simulation platform for generating training and evaluation data sets under multiple noise models, and (3) demonstration that our structured DNN approach improves localization under noise by up to 25% over traditional off-the-shelf DNNs, and can achieve sub-meter accuracy in a real-world experiment.*

## KEYWORDS

*Millimeter wave, 5G, 802.11ad, Localization, Mobile networks, Machine learning, Deep Neural Networks*

## 1. INTRODUCTION

The next generation of wireless networks will utilize high-frequency millimeter waves (mmWaves). A consequence of this is the need for accurate narrow-beam alignment between receiving and transmitting nodes [1]. One such alignment method is to localize mobile nodes such that base station antennas can align their narrow beams (e.g., under 30 degrees) towards mobile nodes, and vice versa, when mobile nodes also possess directional antennas. This need for localization is reflected in the emerging 5G standard, which will use mmWave spectrum and explicitly calls for sub-meter localization accuracy for indoor deployments [2]. We propose new, data-driven localization methods using lower frequency spectrum such as 2.4GHz or 916MHz, as in the experiments in this paper, and their omni- or quasi-directional functionality to provide a low-overhead, accurate localization to be used for aligning higher frequency end points.

While there exist many methods for localization, there are a number of domain-specific challenges related to the goal of providing accurate beam alignment for mmWave wireless networks. First, the localization must be highly accurate if it is to be used for the purposes of aligning narrow beams without significant link loss. Second, as the beam alignment procedure must be executed in settings in which the location of the mobile node is rapidly changing, the localization procedure must be sufficiently fast. Third, the localization procedure must be well suited to the challenges posed by the environments and sensing mechanisms in which the networks will operate. For example, indoor environments are highly susceptible to dynamic multipath effects due to the propagation of electromagnetic waves and their interaction with the high density and diversity of objects and materials within the environments.

To address these challenges, we propose a fully data-driven neural network approach to localizing mobile nodes for the purposes of aligning antennas of communicating devices accurately and in low latency. Our proposed approach is able to localize mobile nodes using signal patterns learned specifically from the data associated with the environment, making our method particularly well suited to complicated multipath environments that are difficult to capture analytically using traditional channel models. Importantly, as our method requires only data to train the localization model, it is not necessary to have any explicit rules or heuristics to function in these complicated environments. For these reasons, our data-driven method is applicable to the highly heterogeneous individual environments in which mmWave systems will be deployed. For a given environment, only a small number of location samples is required (e.g., fewer than 30) for learning the localization neural network. Further, we show that a relatively small deep neural network (DNN) model (e.g., with four layers) is sufficient for accurate localization, and can infer locations in under a millisecond via a feed-forward operation.

We propose novel neural network techniques specifically designed for the high frequency wireless domain and associated antenna alignment application requirements while simultaneously minimizing the amount of real-world training samples that must be collected. Note that in order for such a data-driven approach to be practically applicable, it must be able to operate without requiring overly onerous amounts of data collection for new environments. Further, a fully data-driven approach must automatically address multipath effects, occlusions, and other characteristics of the environment that influence the behavior of electromagnetic waves. Our proposed machine learning algorithms are able to learn these complicated effects automatically.

Our main contributions are as follows:

1. We introduce a structured neural network model to address collinearity regions between base stations with antenna arrays that capture phase differences in received signal from a mobile node to be localized.
2. To minimize the data collection effort, we introduce a simulator for generating training data for the localization neural network under various scenarios including differing number of base stations, number of sample points, and under multiple noise models.
3. We demonstrate with both simulated data and real-world experiments that our structured DNN approach improves localization under all noise models considered by up to 25% over traditional off-the-shelf DNNs. Specifically, we achieve sub-meter in both simulated indoor and real-world experiments.

## 2. RELATED WORK

Our work focuses on localizing mobile nodes using multiple base stations each equipped with antenna arrays in a lower frequency, such as 2.4 GHz. Earlier attempts at localizing nodes with wireless network connections include using signals such as Received Signal Strength Indicator (RSSI) (see, e.g., [16]). RSSI is applicable to nodes using a number of different networking protocols, including Wi-Fi [18] and Bluetooth [17]. However, while RSSI can be used to calculate a coarse estimate of distance between a receiving and transmitting node, in practice the granularity of the localization is not sufficient to meet a goal of sub-meter fidelity [16].

In contrast to directly using RSSI provided by the radio signals, our method uses additional information in signals to localize nodes beyond its signal strength. There are a number of other methods that approach the problem similarly. Adib et al. [4,5] present methods for tracking users through walls using frequency modulated carrier waves (FMCW), or a chirp signal, and multiple users moving within an indoor environment can be localized and tracked. FMCW requires specialized hardware to generate the “chirp” signal that enables the time of flight measurements to be calculated accurately and then used in determining the location. This method relies on the users to be moving in order to separate their reflections from the static background, is limited to just a few users, and requires a windowed set of measurements for accurate background subtraction.

A motivation for our work comes more directly from the 802.11ad Amendment III, which addresses the next generation of indoor wireless networking, focusing on connecting and using high-frequency networks [6]. The amendment proposes to perform beamforming and alignment through a procedure called Sector Level Sweep (SLS), an in-band search to locate the correct beam sectors to utilize for inter-device communication. SLS scans over the available antenna sectors, and is then refined via the Beam Refinement Process (BRP). To make this process more efficient, Blind Beam Steering (BBS) [7] has been proposed to reduce the amount of scanning by utilizing a lower frequency, 2.4 GHz, and computing an angle-of-arrival using MUSIC.

Our approach is similarly motivated by the desire to reduce the amount of scanning by utilizing a lower frequency mode for localization purposes. However, while BBS may need to revert to an SLS when large multipath effects are detected, our data-driven methods instead allow our model to learn multipath effects from both observed and synthetic data, and work effectively and automatically in these regions. Further, by calculating a position rather than just the angle-of-arrival, our method can quickly find one of the alternative beams associated with the location of the mobile node for use during intermittent blockages, doing so without the cost of beam sweeping. In addition, the localization information computed by our method can be useful not only for beam alignment purposes, but also for antenna selection and future medium access control algorithms.

ArrayTrack is a system that utilizes a set of base stations equipped with antenna arrays throughout an environment in order to localize mobile nodes within the environment using angle-of-arrival measurements [8]. This work utilizes as many as eight antennas on each base station in order to limit the error of the MUSIC algorithm when calculating the angle-of-arrival, specifically attempting to improve MUSIC's robustness to the multipath effects plaguing indoor wireless environments.

In the area of using data-driven methods for localization, deep neural networks for performing localization have been previously evaluated under multiple indoor scenarios [9,10,11].

Our method improves upon the localization error presented in the aforementioned work by

modifying the neural network structure (topology) to suit the application domain. Our proposed data-driven method is robust to multipath effects by learning models directly from real measurements that contain these effects. As such, we show our method can provide better accuracy using fewer antennas and base stations in our real-world experiments demonstrating sub-meter fidelity.

### 3. CHALLENGES FOR LOCALIZATION

We illustrate the operating scenario for the overarching localization problem for indoor or outdoor networks operating with dual modes in Figure 1. We consider an environment in which a user is operating a mobile node that may change position over time. The mobile node is served wirelessly by a number of base stations that are dispersed throughout the environment. We assume that each base station contains multiple antennas, with one array operating at out-of-band low frequencies (such as 2.4 GHz in Wi-Fi and 916 MHz in our real-world experiments) and another array operating at higher mmWave frequencies (such as 28, 38, or 60 GHz). (Please note that for the rest of the paper, we will use 2.4 GHz as an example of lower frequencies used for the localization purpose.) In this scenario, it is desirable for the mobile node to communicate with a given base station over the high frequency mmWave channel in order to achieve higher data rates as compared with lower frequency links. However, due to the directional nature of mmWave antennas and systems, it is necessary for the endpoints to know in which direction, or sector, to steer the beam. This can be accomplished directly with knowledge of the location of the mobile node within an environment.

There are a number of challenges associated with localizing mobile nodes within a realistic environment in order to meet specific localization requirements for high data-rate mmWave communications. These challenges stem from both the system itself, such as collinearity problems induced by the placement of base stations, as well as the properties of electromagnetic waves within the environment, such as multipath and fading effects. We now discuss these challenges in detail.

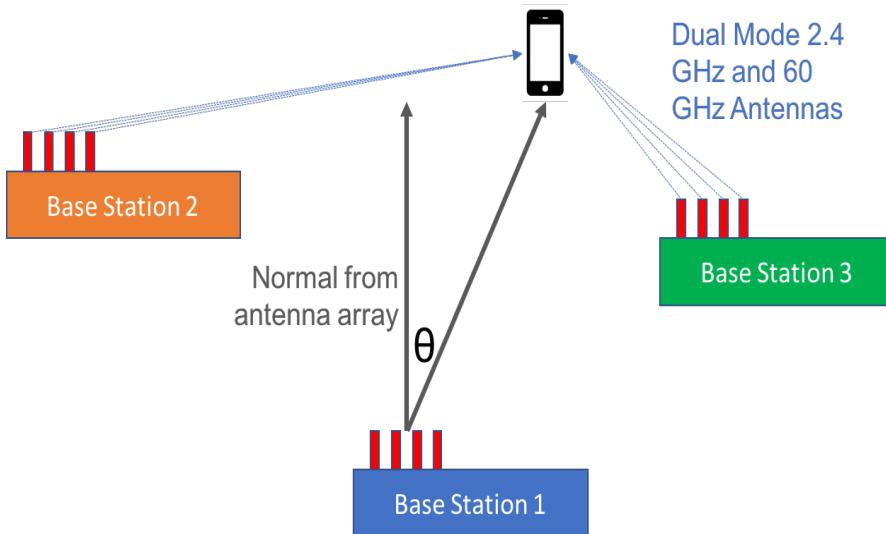


Figure 1: Localization scenario for mmWave networking. In this illustration, we use 60 GHz mmWave for high-speed data transmission and 2.4 GHz frequency band for localization. We also illustrate the manner in which angle-of-arrival is measured, denoted by  $\theta$ .

#### 3.1. Collinearity

Within the context of localizing a mobile node, we define a *collinear region* between a pair of

base stations as the region along a straight line connecting the two base stations, as shown in Figure 2. From the perspective of localizing a mobile node, this collinear region is of special note, as the location of any node located along this line cannot be uniquely determined from the angles-of-arrival between the node and the two base stations or the phase offsets at the two base stations.

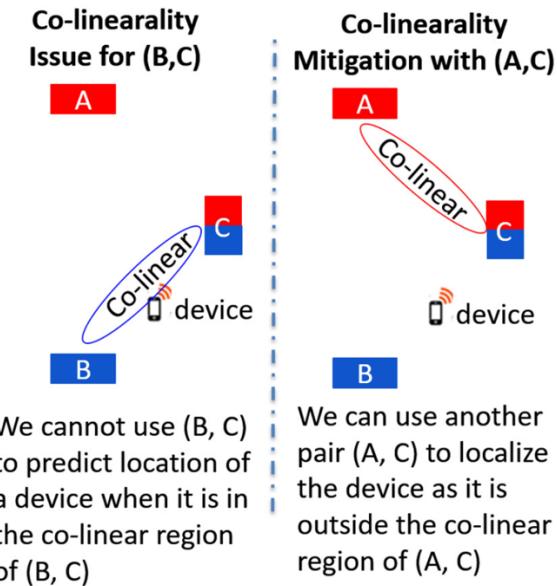


Figure 2: An illustration of the collinearity issue, which the Structured Multi-Layer Perceptron (SMLP) localization neural network of this paper addresses.

In the context of data-driven machine learning, the fact that there exists one input (either angles-of-arrival or phase offsets) that maps to multiple outputs (multiple locations within the collinear region) introduces a one-to-many relationship. This makes it impossible to accurately infer the location of a mobile node correctly within the collinear region. Within the collinear region between two base stations, the model would therefore predict the location incorrectly (i.e., with a large amount of error).

More specifically, the model tends to predict any node within the collinear region as being at the midpoint of the collinear region. This stems from the fact that minimizing the loss for a one-to-many mapping is accomplished by predicting the output as the average of the outputs of all data points whose inputs are the same, which can be shown to be the maximum likelihood estimate. While having greater than two base stations can resolve these regions analytically, data-driven, conventional machine learning models without our proposed modifications to the neural network's structure that reflect the placement of base station nodes have trouble resolving these regions under noise, or require much larger amounts of training data even when noise is not present.

### 3.2. Multipath

In realistic deployment scenarios, the presence of walls, reflective surfaces, objects, and other physical features that interact with the propagation of electromagnetic waves can result in multipath effects. Within the context of localization, multipath interference can often introduce large variations when estimating the angle-of-arrival, as it will alter the phase or time of flight measured at the base stations.

Without accounting for multipath, which can be difficult when building explicit models of the

room or even using statistical models such as MUSIC, the derived location of the mobile node can be very inaccurate. While advancements in these statistical methods have been made to better address this, problems still exist in the presence of large multipath effects [7,8].

For indoor deployments of mmWave networks, it is imperative that the localization methods are able to operate under multipath effects resulting from reflections off of various types of materials. This is primarily due to the great variety of objects, materials, and architectural layouts common to deployment environments. In our proposed data-driven approach, rather than accounting for these effects or attempting to model them explicitly, small amounts of data can be sampled from the room and used to train a model that captures the key propagation characteristics of the given scenario. As long as regions of the environment with large multipath effects are sampled, these key characteristics can be learned by the model.

### 3.3. Latency and Interference

The procedure for selecting which beam sector to use, or beam to form, may be executed in settings in which the location of the mobile node is rapidly changing. As a result, any localization procedure must be sufficiently fast to meet these conditions. When the alignment latency between end points is too long, the system initialization overhead will be high, while intermittent latency (e.g., MAC ARQ) resulting from packet loss due to misaligned beams can potentially disrupt higher layer protocols, causing loss of throughput from, for example, TCP timeouts [12]. Further, as many new applications have stringent latency budgets, such as Virtual Reality (VR) and Augmented Reality (AR), it is important that deployments requiring localization are cognizant of the latency cost of the localization method used.

Some of the approaches to beamforming are particularly not well suited to the stringent latency budgets of emerging applications. For example, blind in-band beam scanning in high frequencies, in which the optimal antenna sector is found by scanning over many antenna sectors, incurs delays that may make it incompatible with use in low-latency systems. In addition, this scanning procedure can introduce interference, especially under multi-mobile node and multi-base station scenarios, which further reduces its practicality.

In contrast to this, deep neural networks have recently seen significant improvement in feedforward inference time and storage costs. For example, binarized neural networks, where all the weights are 1-bit values, can be used even on memory and power constrained embedded devices [13,14]. For such neural networks, the computation time for networks with fewer than ten layers and small input sizes can be performed extremely efficiently and with latency under a millisecond, as discussed in Section 5. As such, our proposed data-driven methods are cognizant of the latency requirements of emerging applications.

Further, as our methods localize the node using a separate, lower frequency band rather than the mmWave bands used for high speed data transmission, our method does not suffer from the interference introduced by the beam scanning procedure. As such, our method can function without interfering with other mobile nodes and base stations operating in the same environment, as is expected to be the case in many mmWave deployments.

## 4. DATA-DRIVEN LOCALIZATION METHODS

We now describe the specifics of our data-driven DNN-based approach for predicting the location of a mobile node given measurement data at the base stations. We name our method the Structured Multi-Layer Perceptron (SMLP). The SMLP is a structured learner and inference engine consisting of multiple deep neural networks (which we define as neural networks with more than one hidden layer). The SMLP, as we will show, improves the localization accuracy and robustness for localization within collinear regions beyond the capabilities of off-the-shelf

DNNs.

#### 4.1. The Structured Multilayer Perceptron Model

The Structured Multilayer Perceptron model (SMLP) is a type of DNN with domain specific neural network modifications. At a high level, SMLPs are a combination of multiple DNNs structured in a way such that the model can automatically address the challenge of localizing mobile nodes within regions of collinearity. Without these modifications, conventional neural networks such as a Multi-layer Perceptron (MLP) would find it difficult to accurately localize mobile nodes in collinear regions without large amounts of training data.

We now detail the construction of the SMLP, which is illustrated in Figure 3. The SMLP is constructed of two components, a **Lower Pair Network** (LPN) and an **Upper Connection Network** (UCN), which both may contain a number of MLPs. Each MLP, both within the LPN and the UCN, is formulated as a *regression* problem. Specifically, the output of each MLP is an unrestricted physical location in the real space, which consists of all possible locations. This is in contrast to formulating the training as a *classification* problem, which would instead have as output a location chosen from a predetermined and finite set of locations, or bins.

The LPN consists of a combination of submodels, each of which are fully connected MLPs. Each neural network in the LPN is trained on the data from pairs of base stations with collinear regions. Importantly, each neural network in the LPN is trained independently of one another. Once trained, each neural network in the LPN has the functionality to be able to take as input the angle-of-arrivals to each of the base stations, and output a localization of the mobile node.

Note however, that due to the way each network is trained, these individual networks will not be able to accurately localize nodes within the collinear region. This motivates the introduction of the UCN, which takes the output from all of the LPN networks and uses it to resolve the collinear problems. Specifically, we remove the final output layer of each network in the LPN, and concatenate the output together. This is then used as the input to the UCN, which outputs the final localization of the mobile node. By having a global view from multiple base station pair combinations by proxy from the concatenated LPN output, the UCN is able to rectify the collinearity problem, often with significantly less training examples.

#### 4.2. Training SMLPs

We now discuss the training procedure for SMLPs for the wireless localization scenario. SMLPs are trained offline on either collected or synthetic training data. On a 3.4 GHz CPU with 16GB of RAM, the models used in all results in the paper can be trained in less than one minute. As discussed previously, all training is formulated as a regression problem rather than a classification problem, such that the final output of the SMLP is a location within the real space rather than a probability distribution over predetermined locations.

All portions of the SMLP are trained using conventional neural network training methods, which we now briefly describe. To train a fully connected neural network with  $x$  hidden layers and  $[x_1, x_2, \dots x_{n-1}, x_n]$  neurons per hidden layer, initial weights are assigned to each neuron in each layer. Paired (input, output) training samples are then used to find the optimal weights for each neuron. Each input, either individually or in a batch, is fed through the neural network, producing a final output. This final output is then compared to the ground truth output of the training sample via a loss function. For our models in this paper, we use an  $l_2$ -norm loss function, such that the loss function calculates the square-root of the squared difference between the output of the neural network and the ground truth output. This error is then propagated backwards through the network using *backpropagation*, a process through which the weights of each of the neurons is adjusted conditional on the amount of error. This process is repeated

until convergence or until a user-defined number of iterations is reached.

We now discuss the training process for the Lower Pair Network (LPN), and afterwards describe the training process for the Upper Connection Network (UCN). The training samples for each neural network in the LPN are angles-of-arrivals (AOAs), as measured by the corresponding pairs of base stations. The output used in training the model is the location of the mobile node in the real space. We use a rectified linear unit (ReLU) as the activation function, and an  $l_2$ -norm loss function. Each neural network within the LPN is trained independently of one another, as there are no connections between each unit of the LPN, only in the UCN.

We now describe the training process for the UCN. The input to the UCN training is derived from the output of the LPN. Specifically, the last layer from each of the neural networks within the LPN is removed. Following this, the output from the new final layer from each of the neural networks in the LPN are concatenated to form a single long vector. This vector is used as input in training the UCN. The output used in training the UCN is the location of the mobile node in the real space. We again use a ReLU as the activation function, and an  $l_2$ -norm loss function.

Note that each neural network within the LPN as well as the UCN are all trained independently of one another. Specifically, any error in training the UCN is not back propagated to the LPN. By combining the output of the multiple complementary LPN models at the UCN, the SMLP is able to resolve the collinear regions that the lower layer submodels are unable to resolve by themselves. As such, there is no need to backpropagate through the entire model, as the UCN resolves the errors made in collinear regions by the LPN.

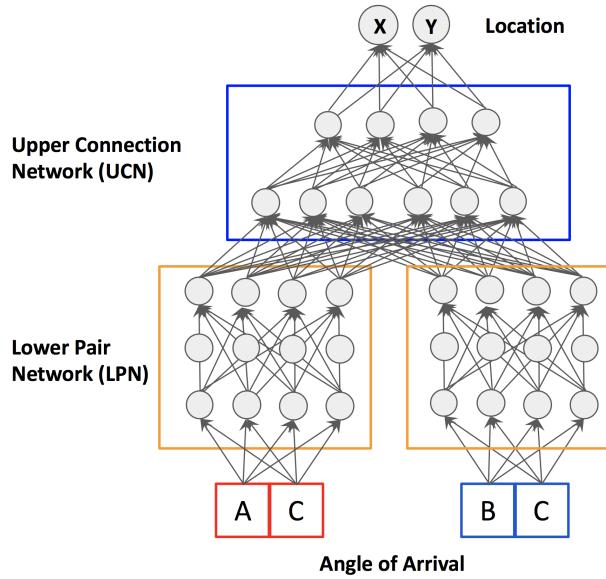


Figure 3: A diagram of our proposed Structured Multilayer Perceptron (SMLP) model, which separates base stations pairs in the Lower Pair Network (LPN) and then combines them in the Upper Connection Network (UCN). The SMLP infers the  $(x, y)$ -coordinates of a mobile device.

## 5. SIMULATION AND RESULTS

In order to validate our proposed methods, we have built a detailed and realistic simulator, and use it to generate data with which our methods can be evaluated. We first describe the simulator we have built for these purposes. We next present results using the simulator under a number of different conditions, including complex noise models validated through real-world experiments. For all results in this section, we use a virtual setup with three base stations, and evaluate on a

held-out test set of 2500 points canvassing the entire 50x50 virtual space corresponding to a 3m by 3m grid. All results are reported in terms of Median Square Error (Median SE).

Before describing results, we briefly comment on the latency ramifications of SMLPs. Note that the SMLP models in this section use 2 layers in the Lower Pair Network (LPN) with 500 and 50 neurons per layer, respectively, and 2 layers in the Upper Connection Network with 200 and 50 neurons per layer, respectively. Further, the number of multiply and addition operations needed to use the SMLP for inference can be found according to the following formula:

$$Ld_l n_{l1} n_{l2} + U d_u n_{u1} n_{u2} n_{uo}$$

$L$  - # of LPNs

$d_l$  - dimension of LPN input

$n_{lk}$  - # of neurons in LPN layer  $k$

$U$  - # of UCNs

$d_u$  - dimension of UCN input

$n_{uk}$  - # of neurons in UCN layer  $k$

Therefore, for the SMLP used in this paper, a total of

$$2 \cdot 3 \cdot 500 \cdot 50 + 1 \cdot 50 \cdot 200 \cdot 50 = 650,000$$

multiply and addition operations are required. A commodity accelerator capable of executing 4500 GFLOPs can execute these operations in 0.01 milliseconds, making our method applicable in scenarios with small latency budgets.

### 5.1. Simulator

In order to evaluate our proposed SMLP Model, we have created a simulator for generating training data under a number of different scenarios of interest, including equipment failure and noise. We have released the simulator publicly as a community resource. The code for the simulator can be found online and is under active development (currently available at <https://github.com/KevinHCChen/wireless-aoa/> ).

The simulator first constructs a configurable virtual space in which the experiment will exist. The simulator then places mobile nodes within the space, as well as base stations. Given this setup, the simulator generates the data needed to train the SMLP models. Specifically, based on the relative location of the base stations and the mobile nodes, the simulator generates either time of flight or angle-of-arrival data to each of the base stations given a virtual mobile node and its position, and pairs this with the location of the mobile nodes. This data is then exported such that it can be used in training the SMLP models.

The simulator can be easily configured via initialization files, whose parameters include:

1. Distribution of mobile locations (uniform grid or random)
2. Number of mobile locations
3. Dimensionality of simulation (2D or 3D)
4. Number and location of base stations

In order to simulate realistic conditions that real-world deployments will face, we also incorporate several noise and filtering modules to emulate fading and multipath effects. These

include:

1. Gaussian noise
2. Angle Dependent Noise
3. Base station outages
4. Uniform noise

We describe these noise models further in Section 5.3 below.

## 5.2. Collinearity Mitigation

In this section, we demonstrate that SMLPs are able to rectify localization errors within collinear regions. As described in Section 3.1, the presence of collinear regions between base stations, even when a sufficient number of base stations are used, introduce difficulties for accurate localization of mobile nodes.

We train an SMLP model with 2 layers in the Lower Pair Network (LPN) with 500 and 50 neurons per layer, respectively, and 2 layers in the Upper Connection Network with 200 and 50 neurons per layer, respectively. As a point of comparison, we train a conventional (not-structured) fully connected MLP model with 4 layers and 500, 50, 200, 50 neurons per layer, respectively. Figure 4 shows the error in localizing points within the simulated space with three base stations for the MLP and SMLP models, where the error is defined in meters between the predicted and true locations of each mobile node. We find that while the MLP model has large amounts of error in the collinear regions between base stations (denoted by the red regions in Figure 4 on the left), the SMLP model is able to resolve these regions with errors comparable to those in non-collinear regions (as seen in the blue regions in Figure 4 on the right). Note that in order to make a fairer comparison, we additionally train a not-structured MLP with more neurons per layer such that the total number of neurons is comparable to that of the SMLP, and find that this does not have a marked impact on improving its performance.

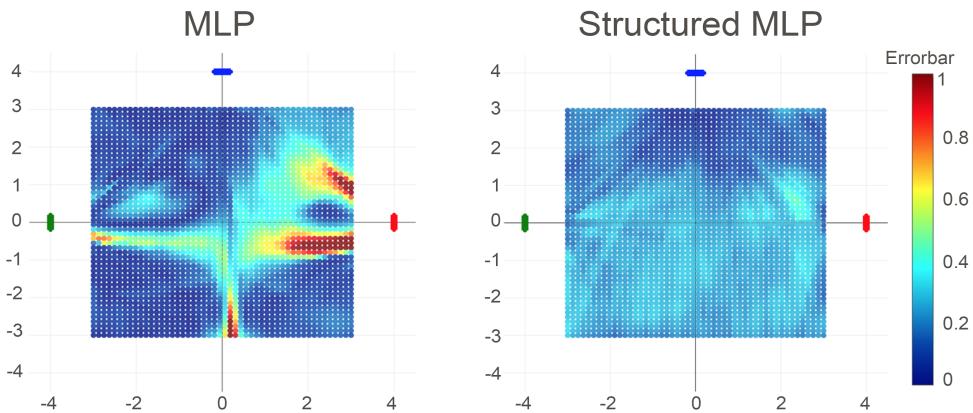


Figure 4: When simulating data without noise, the non-structured MLP (left) is unable to resolve the collinear regions while our proposed Structured MLP (right) model can.

We next examine the Median SE achieved by each of the models in localizing all 2500 data points within the simulated space. Figure 5 shows the Median SE achieved by the SMLP and non-structured MLP models for different amounts of training data, ranging from 50 to 2000 points. We find that for any amount of training data used, the SMLP outperforms the MLP in terms of Median SE.

Intuitively, this is due to the fact that given two points in different positions but within the same

collinear region, the UCN in the SMLP is able to use the output from the two LCN submodels to resolve the collinear region problem, while the MLP, even with the same number of layers, is not able to do so.

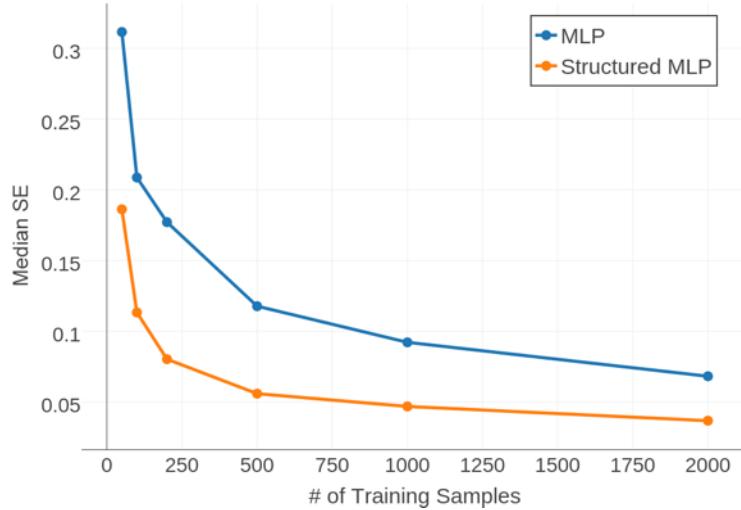


Figure 5: Using simulation data without noise, we show the improvement in localization accuracy, in Median SE, of the Structured MLP (SMLP) as compared to the non-structured MLP model. Further, we find that for a given accuracy, the SMLP can localize nodes using only a fraction of training data required by the non-structured MLP for comparable performance.

Further, in real-world deployments in which data must be collected, it is of interest to be able to train the model with as few training points as possible while retaining its ability to localize a mobile node to high accuracy, such as sub-meter. At all amounts of training data in Figure 5, the amount of error is far less than one meter. Importantly, we find that with as few as 50 training points, the localization is only 0.40 meters (the Median SE is 0.16 meters).

### 5.3. Resilience Against Noise

We now present results in the presence of noise. In real-world deployments, there are a number of different forms of noise that will be present in the data. To evaluate the performance of our proposed localization methods under various amounts of noise in a consistent and repeatable way, we include several noise models within the simulator and test the models under identical conditions. We concentrate on two main types of noise, Additive Gaussian White Noise (AGWN) and Angle Dependent Noise.

The first type of noise, AGWN, represents the type of noise that arises from antenna movement, inaccuracies in measurement data, equipment mis-calibrations, and interference. To generate the AGWN data, we sample  $n$  points from a Gaussian distribution with mean  $\mu$  and standard deviation  $\sigma$ , where  $n$  is the number of data points in the dataset, and then add it to the generated input data. In experiments, we use  $\mu=0$  so the noise is centered around the true value and vary the value of  $\sigma$ .

The second type of noise is Angle Dependent Noise. The Angle Dependent Noise models the fact that measurements become noisier as the angle between the normal from the antenna array on the base station and the mobile node increases, denoted by  $\theta$  in Figure 1. More specifically, when the mobile node is aligned in front of the base station antenna (i.e., close to the normal

from the antenna array), there is little noise compared with when the mobile node is oriented, for example, near 90 degrees from the normal of the antenna array. Accordingly, we use a nonlinearity function  $a(\theta)$  to model the effect of the noise, as defined in Equation 1, where angle  $\theta$  is defined in Figure 1,  $k$  controls for the magnitude of the nonlinearity, and  $j$  controls for the base amount of noise (i.e., noise when the angle is 0 degrees).

$$(1) \quad a(\theta) = e^{|k(\theta-90)|} - j$$

This model reflects what we find through empirical observation in the field with equipment.

In generating data under this noise model, we first generate noise from a Gaussian distribution and then multiply it by the nonlinear factor  $a(\theta)$  according to the location of the point, as shown in Equation 2.

$$(2) \quad a(\theta) \times N(\mu = 0, \sigma)$$

Just as in the case of the AGWN model, the Angle Dependent Noise model can be automatically added and configured with different parameters for use by the simulator. Now that we have described the two noise models, we examine how our data-driven methods operate in the presence of noise.

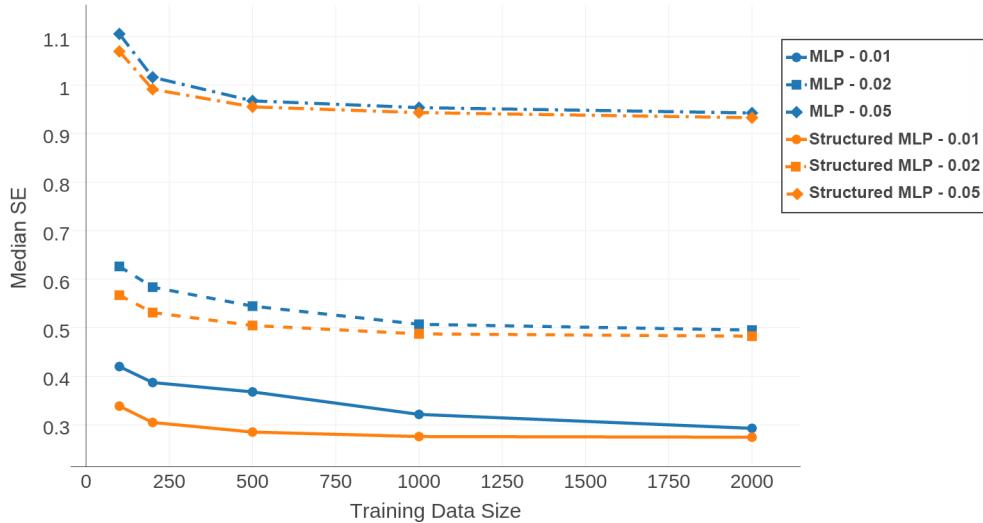


Figure 6: Using simulation data with Additive Gaussian White Noise (AGWN), we show the improvement in localization accuracy of the Structured MLP (SMLP) as compared to the non-structured MLP model.

We first examine the impact on performance under the AGWN model. Figure 6 shows the impact of the AGWN model on both the proposed SMLP method and the non-structured MLP for different amounts of noise and different amounts of training data, where the amount of training data is the number of sampled locations. The noise is sampled from Gaussian distributions with  $\mu=0$  and  $\sigma=\{0.01, 0.02, 0.05\}$ . These standard deviation settings translate to average amounts of angular noise of 2.9 degrees, 5.7 degrees, and 14.4 degrees, respectively. We find that the model is robust to small to medium amounts of Gaussian noise, both achieving accuracies well within the 1 meter fidelity goal. Further, we find that the SMLP significantly outperforms the non-structured MLP at all settings of noise and amounts of training data, with the gap between the models increasing as  $\sigma$  decreases. Finally, we find that at even high levels

of noise, given a sufficient amount of training data (250 points), the SMLP model is able to meet the 1 meter fidelity goal, and with smaller amounts of training data, performance suffers only marginally (error rising from 0.99 median meters at 250 training points to 1.03 median meters with only 100 training points). Importantly, this shows that even under a significant amount of noise, the SMLP model can still approximately meet the 1 meter fidelity goal with little data.

We next examine how the addition of noise impacts how much data is needed to train the model. Figure 6 shows the impact of Gaussian noise as the number of training samples is varied for different levels of noise. We find that at small to medium levels of noise, even with as few as 100 points, the 1 meter fidelity goal is easily reached (with a localization error of 0.58 median meters and 0.75 median meters for  $\sigma = 0.01$  and  $0.02$ , respectively).

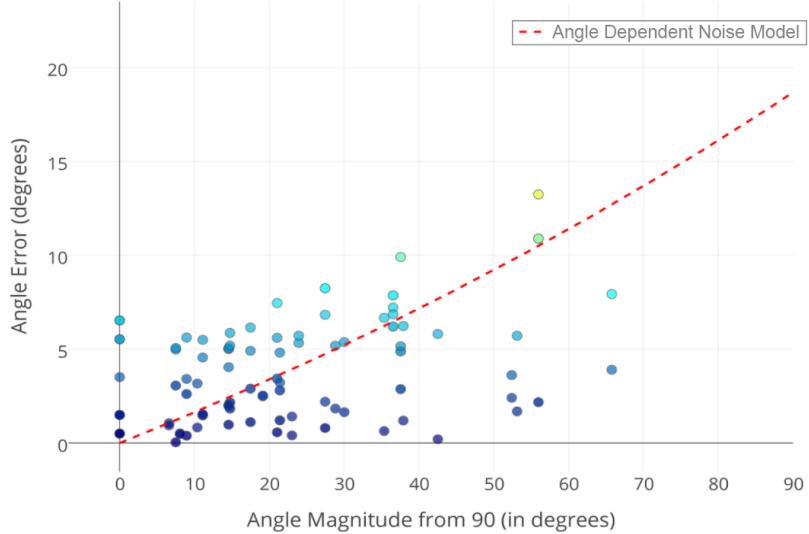


Figure 7: Our Angle Dependent noise model  $a(\theta)$  with  $k=1$  and  $j=4$  (red dotted line) reflects real-world data captured from field experiments (points) over various angles (x-axis), as shown by the fact that the noise model mirrors the real-world points as the angle varies.

We now examine the impact of adding Angle Dependent Noise. For these experiments, we set the parameters as  $k=1$  and  $j=4$  in the nonlinearity defined in Equation 1. These parameters are set based on empirical observation of field experiments performed with a set of N200 User Software Radio Peripheral (USRP) software defined radios, using the setup as described in Section 6 to measure the amount of Angle Dependent Noise present in a true system deployment. This allows us to properly set the parameters in the nonlinearity. Figure 7 shows the nonlinearity defined in Equation 1 with these parameters superimposed on top of real data collected in the field, where the x-axis shows the true angle offset between the mobile node and the normal from the antenna array, and the y-axis show the amount of error in measuring the angle-of-arrival to that point. The figure shows that the nonlinearity model tracks real-world measurements under these parameter settings.

Given these parameters, we simulate Angle Dependent Noise for varying levels of Gaussian noise,  $\sigma=\{0.01, 0.02, 0.05\}$ , multiplied by the nonlinearity defined in Equation 1. Figure 8 shows the impact of Angle Dependent Noise for different amounts of Gaussian noise and varying amounts of training data. Overall, we find that the amount of localization error under the Angle Dependent Noise model is less than that under the AGWN model. This can be explained by the fact that the Angle Dependent Noise model largely only affects those points at a greater angular offset from the normal from the antenna and the AGWN model affects all

points equally. We find that the model is able to achieve the goal of at least 1 meter fidelity with small, medium, and large amounts of Gaussian noise under the Angle Dependent Noise model. Further, the SMLP significantly outperforms the MLP with small and medium amounts of noise at all amounts of training data, with the gap between the models again increasing as  $\sigma$  decreases. Interestingly, the SMLP model at  $\sigma=0.02$  is able to achieve approximately the same error as the non-structured MLP model at  $\sigma=0.01$  with small amounts of training data. This shows that under the Angle Dependent Noise model, the SMLP model can handle up to twice the amount of noise as the MLP model.

We also examine how the addition of noise impacts how much data is needed to train the model. Figure 8 shows the impact of Gaussian noise under the angle dependent model as the number of training samples is varied for different levels of noise. We find that at all levels of simulated noise, even with as few as 100 points, the 1 meter fidelity goal is easily reached (with a localization error of 0.51m, 0.65m, and 0.88m, for  $\sigma = 0.01$ , 0.02, and 0.05, respectively, all well within the 1 meter goal).

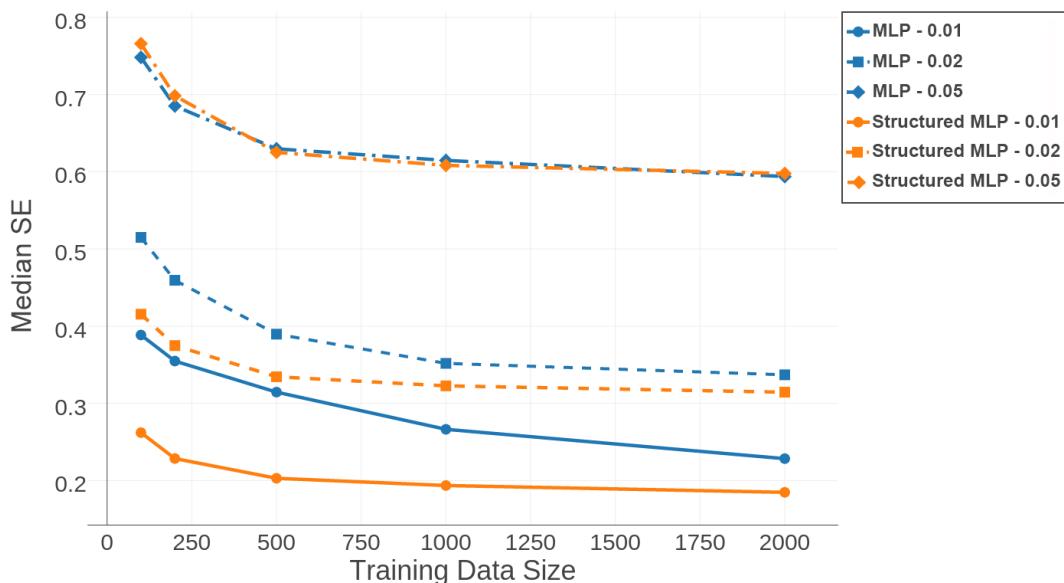


Figure 8: Using simulation data with Angle Dependent Noise, we show the improvement in localization accuracy of the Structured MLP (SMLP) as compared to the non-structured MLP model.

#### 5.4. 3D Simulations

For completeness, we also include a 3D component to our simulator in order to verify our models with the added dimensionality. In practice, to capture a 3D position, the base stations must have antennas in two polarities, usually vertical and horizontal. Our simulator generates positions within the 3D space, and then calculates angle-of-arrival or time-of-flight for each polarity. From this, we can evaluate our models for 3D data in the same manner as we do with 2D data.

We find that the 3D results follow the same trends presented in the previous results section. This is due to the fact that the primary difference between MLP and SMLP models is their ability to address the collinear region, which remains roughly the same in size in both the 2D and 3D cases, as the region lies within the 2D plane with minimal impact in the added third dimension. Figure 9 is a 3D rendering of the predicted positions and associated Median SE

from two different viewpoints: a top-corner view and a ceiling view. As previously seen for the 2D results, the error concentrates itself along the collinear region along the straight line between base stations. We have omitted complete results for the 3D simulations due to space concerns, as they very closely mirror that of the 2D results.

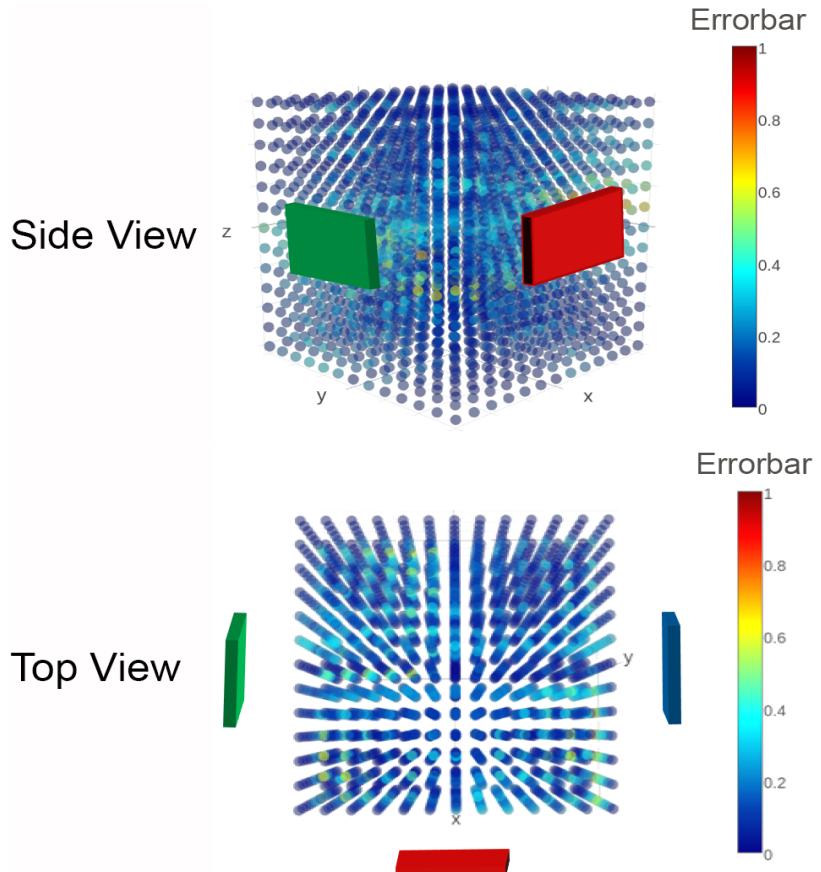


Figure 9: Our simulator is capable of generating and testing our localization methods under 2D and 3D environments, as shown in the two 3D views above.

## 6. REAL-WORLD EXPERIMENTS

In addition to the simulated experiments discussed in the previous section, in this section we present a real-world experiment performed in a realistic deployment scenario. We first describe the experimental equipment. Our experimental setup consists of a collection of receivers, a transmitter, and a reference node. We use N200 User Software Radio Peripheral (USRP) software defined radios operating at 916 MHz. The receiver USRPs are connected to a portable Dell Inspiron 410 via a Netgear gigabit switch, and serves as a control node responsible for starting experiments and collecting measurements from the USRPs. The transmitter sends a predefined symbol string of a thousand 1s that is captured by all of the receiver USRPs. The reference point transmits the same symbol string and is placed directly in front of the base station so the phase offsets can be used to calibrate the transmitter values. We follow the same experimental approach and reference node implementation described in [15].

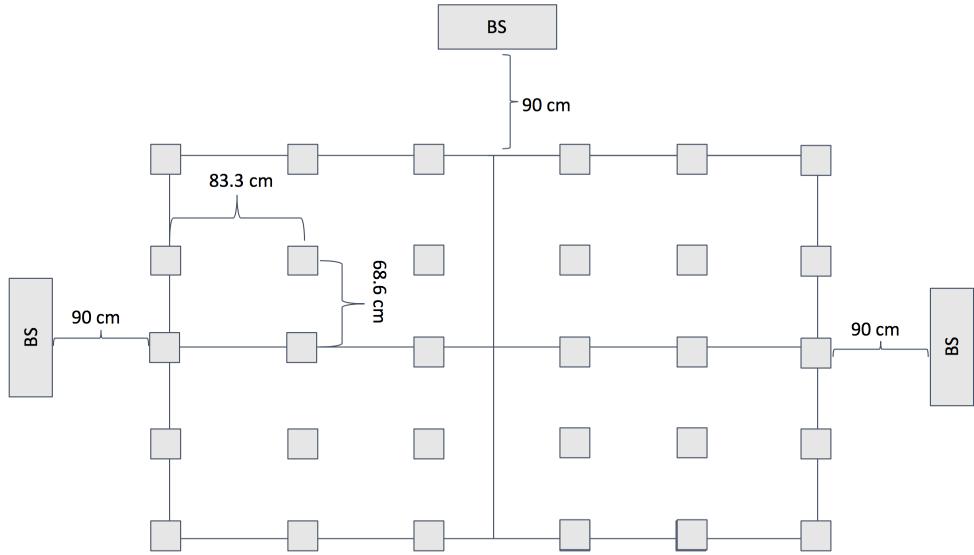


Figure 10: Layout of indoor and outdoor experimental setups: a five by six grid for the transmitting USRP location (for the mobile node) with three base station (BS) locations around the grid

In order to evaluate our models on real-world datasets, we use the setup shown in Figure 10. We use a five by six grid of location points, resulting in 30 locations in total. We use three base station locations. However, due to equipment limitations (having only one base station node consisting of four USRPs as an antenna array of four antennas), we move the base station to the three locations sequentially and re-measure at each location, but logically this is equivalent to having three base stations measuring at once. Note that we use a two-dimensional setup, as we lack an antenna array with vertical polarity. However, based on the similarity between our 2D and 3D simulation results, we expect that 3D results in the field would be very similar to the results presented here. The environment is an outside, open, grassy area, as shown in Figure 11.



Figure 11: The real-world experiment environment

For the experiment, we collect 90 samples (3 samples at each of the 30 points on the grid) per

day for two days. The model is trained on data from one day, and tested on data from the other day, where the second day is completely held out from training (i.e., the model is never trained and tested on data collected on the same day). We find that the SMLP achieves a Median SE of 0.16 meters (0.4 meter error) in this experiment, achieving sub-meter accuracy, while the unstructured DNN performs worse, and achieves a Median SE of 0.2 meters (0.45 meter error).

## 7. IMPACT

In this section, we describe the impact of our work presented in this paper. As shown in Sections 5 and 6, our proposed methods are able to achieve sub-meter localization accuracy for mobile nodes, even in the presence of noise. mmWave transmitters and receivers will likely need to align their beams via a beam forming process, as the alignment accuracy between the transmitter and receiver will determine the quality of the signal between the two. However, the exact impact of localization accuracy on signal quality, as well as baseline levels of localization accuracy needed for high-rate mmWave communication, is dependent on antenna choices, specifically the choice of omni or directional antennas on base stations and mobile nodes, and the half power beam width of the antennas. As such, we have set the 1 meter indoor localization fidelity goal in the 5G standard [2] as our goal in developing the methods. In this section, we go beyond this exact numerical goal, and describe the potential impact of our method by considering two possible systems and the ramifications of our localization on these systems.

Specifically, we consider setups in which the antenna and equipment choices result in 1) the existence of an exact beam alignment that allows for higher signal quality than any other alignment choice (i.e., there exists one alignment, such as an exact alignment between the transmitter and receiver, that is strictly better than all other alignments), and 2) the existence of a window of alignments such that as long as the alignment of the transmitter and receiver is within this window, signal quality is identical or near identical (i.e., for a given distance, as long as the alignment is within some  $x$  amount of error from the true alignment, there is no impact on signal quality, where  $x$  will decrease as distance between the transmitter and receiver increases). Note that scenario one would arise when using highly directional antennas with small half power beam widths, while scenario two would arise when using an omni-directional antenna for either the transmitter or receiver, or using directional antennas with relatively large half power beam widths.

Under scenario one, our improved localization accuracy will strictly improve the signal quality between the transmitter and receiver. As there exists an optimal alignment between the transmitter and receiver, which will commonly correspond to the line-of-sight alignment between the transmitter and receiver, improved localization accuracy will allow for an as near to optimal alignment as is possible, subject only to localization error. As such, under this scenario, our improved localization accuracy will improve signal quality as compared with less accurate methods.

Under scenario two, for a given distance between a transmitter and receiver, our improved localization method will only improve upon other methods in situations in which other localization methods are not sufficiently accurate as to allow for alignment within the error window. Beyond this however, our method does offer significant gains under scenario two when the distance between the transmitter and receiver increases. As this distance between the transmitter and receiver increases, the size of the error window will decrease. In this respect, the more accurate localization offered by our method will allow for greater distances between the transmitter and receiver to be used without affecting signal quality due to misalignment. (However, signal attenuation and other effects may still impact performance as distance increases).

## 8. CONCLUSION

In this paper, we have introduced Structured Multilayer Perceptrons (SMLPs), a data-driven deep neural network (DNN) localization approach to enable precise narrow beam alignment required by mmWave to deliver data at 10Gb/s or higher rates. Our indoor simulations and real world experiment outdoors demonstrate that we can achieve sub-meter localization accuracy. In general, the localization capabilities of this paper will allow other layers in the network stack to provide improved access control, and therefore improve overall system throughput. Future work will focus on more real-world experiments using these low frequency localization methods, as well as integrating these capabilities into the networking stack in order to take advantage of accurate location information to power smarter, faster next-generation wireless networks.

## ACKNOWLEDGEMENTS

This work is supported in part by gifts from the Intel Corporation and in part by the Naval Supply Systems Command award under the Naval Postgraduate School Agreements No. N00244-15-0050 and No. N00244-16-1-0018.

## REFERENCES

- [1] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "IEEE 802.11ad: directional 60 ghz communication for multi-gigabit-per-second Wi-Fi [invited paper]," IEEE Communications Magazine, vol. 52, no. 12, pp. 132–141, 2014.
- [2] N. Alliance, "5g white paper," Next Generation Mobile Networks, White paper, 2015.
- [3] R. Schmidt. "Multiple emitter location and signal parameter estimation," IEEE transactions on antennas and propagation vo. 3, no. 3, pp. 276-280, 1986.
- [4] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person localization via rf body reflections," in 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15), pp. 279–292, 2015.
- [5] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3d tracking via body radio reflections," in NSDI, vol. 14, pp. 317–329, 2014.
- [6] I. . working group et al., "Ieee 802.11ad, amendment 3: Enhancements for very high throughput in the 60 ghz band," 2012.
- [7] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with eyes closed: mm-wave beam steering without in-band measurement," in 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 2416–2424, IEEE, 2015.
- [8] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in NSDI, pp. 71–84, 2013.
- [9] G. Félix, M. Siller, and E. N. Álvarez. "A fingerprinting indoor localization algorithm based deep learning," Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference on. IEEE, 2016.
- [10] X. Wang, L. Gao, S. Mao, and S. Pandey. "CSI-based fingerprinting for indoor localization: A deep learning approach," IEEE Transactions on Vehicular Technology vol 66, no. 1, pp. 763-776, 2017.
- [11] J. Luo and H. Gao. "Deep belief networks for fingerprinting indoor localization using ultrawideband technology," International Journal of Distributed Sensor Networks, 2016.
- [12] S. J. Tarsa, M. Comiter, M. B. Crouse, B. McDanel, and H. Kung, "Taming wireless fluctuations by predictive queuing using a sparse-coding link-state model," in Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networkingand Computing, pp. 287–296, ACM, 2015.
- [13] M. Courbariaux, Y. Bengio, and J.-P. David, "Binary connect: Training deep neural networks

- with binary weights during propagations," in Advances in Neural Information Processing Systems, pp. 3123–3131, 2015.
- [14] B. McDanel, S. Teerapittayanon, and H. Kung, "Embedded binarized neural networks," in 2017 International Conference on Embedded Wireless Systems and Networks (EWSN 2017), 2017.
  - [15] H.C. Chen, T.H. Lin, H. Kung, C.K. Lin, and Y. Gwon, "Determining rf angle of arrival using cots antenna arrays: a field evaluation," in Military Communications Conference, 2012-MILCOM 2012, pp. 1–6, IEEE, 2012.
  - [16] R. M. Vaghefi, R. G. Mohammad, R. M. Buehrer, and E. G. Strom, "Cooperative received signal strength-based sensor localization with unknown transmit powers," in IEEE Transactions on Signal Processing 61, no. 6, pp. 1389-1403, 2013.
  - [17] M. E. Rida, F. Liu, Y. Jadi, A. A. Algawhari, and A. Askourih, "Indoor location position based on bluetooth signal strength," in Information Science and Control Engineering (ICISCE) 2nd International Conference 2015, pp. 769-773, IEEE, 2015.
  - [18] S. He and S-H. G. Chan, "Sectjunction: Wi-Fi indoor localization based on junction of signal sectors," in Communications (ICC), 2014 IEEE International Conference, pp. 2605-2610, IEEE, 2014.

### Authors

Marcus Comiter is a Ph.D. candidate in Computer Science at Harvard University. He received his A.B. in Computer Science and Statistics from Harvard University.



Michael Crouse is a Ph.D. candidate in Computer Science at Harvard University. He received his B.S. and M.S. in Computer Science from Wake Forest University.



HT Kung is the William H. Gates Professor of Computer Science and Electrical Engineering at Harvard University. He is interested in computing, communications and sensing. Prior to joining Harvard in 1992, he taught at Carnegie Mellon for 19 years after receiving his Ph.D. there. Professor Kung is best known for his pioneering work on I/O complexity in computing theory, systolic arrays in parallel processing, and optimistic concurrency control in database systems. His academic honors include Member of National Academy of Engineering and the ACM SIGOPS 2015 Hall of Fame Award (with John Robinson) that recognizes the most influential Operating Systems papers that were published at least ten years in the past.

