2023 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) | 979-8-3503-0052-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SECON58729.2023.10287483

Learning-based Techniques for Transmitter Localization: A Case Study on Model Robustness

Frost Mitchell
University of Utah
Salt Lake City, USA
frost.mitchell@utah.edu

Neal Patwari
Washington University in St. Louis
St. Louis, USA

Aditya Bhaskara
Sneha Kumar Kasera
University of Utah
Salt Lake City, USA

***Abstract*—Transmitter localization remains a challenging problem in large-scale outdoor environments, especially when transmitters and receivers are allowed to be mobile. We consider localization in the context of a Radio Dynamic Zone (RDZ), a proposed experimental platform where researchers can deploy experimental devices, waveforms, or wireless networks. Wireless users outside an RDZ must be protected from harmful interference coming from sources inside the RDZ. In this setting, localizing transmitters that are causing interference is critical. One notable obstacle for developing data-driven methods for localization is the lack of large-scale training datasets.**

As our first contribution, we present a new dataset for localization, captured at 462.7 MHz in a 4 sq. km outdoor area with 29 different receivers and over 4,500 unique transmitter locations. Receivers are both mobile and stationary, and heterogeneous in terms of hardware, placement, and gain settings. Next, we propose a new machine learning-based localization method that can handle inputs from uncalibrated, heterogeneous receivers. Finally, we leverage our new dataset to study the *robustness* of our technique and others against "out of distribution" (OOD) inputs that are common in most real life applications. We show that our technique, CUTL (Calibrated U-Net Transmitter Localization), is 49% more accurate on in-distribution data, and more robust than previous methods on OOD data.

***Index Terms*—transmitter localization, model robustness, RF spectrum sensing**

I. Introduction

Localization is a classic problem, fundamental to providing ubiquitous mobile and wireless services. More recently, transmitter localization has been used to identify sources of wireless interference [1], [2]. As wireless spectrum becomes more and more saturated, interference is an increasing problem, and novel frameworks for spectrum sharing and management are required to allow for effective spectrum use. One such framework is a Radio Dynamic Zone (RDZ), recently proposed by the National Science Foundation and discussed in recent works [3], [4]. An RDZ is envisioned as a geographic area with greater flexibility in spectrum allocation and usage than currently exists in today's licensed systems. These zones are planned as experimental platforms where researchers deploy experimental devices, waveforms, or wireless networks. As a framework, RDZs will require a system that automatically manages spectrum usage and protects outside users from harmful interference. In this setting, localizing transmitters that are causing interference is critical.

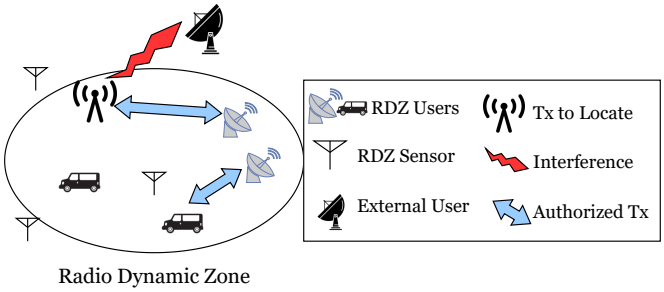

Fig. 1. Locating a transmitter in a Radio Dynamic Zone is a critical part of operating a spectrum sharing system.

Fig. 1 shows an example of localization in an RDZ setting. One transmitter is causing interference to a user outside the zone, so data from the "RDZ Sensors" must be used to localize the interfering transmitter. In order to provide greater coverage, the management system can crowdsource additional measurements from experimental users to more accurately locate a transmitter. In this setting, sensors may be fixed or mobile, and we assume transmitters have full mobility. Although we view transmitter localization in the context of an RDZ, all of the techniques in this work broadly apply to localization of mobile transmitters in other settings as well.

In this paper, we use received signal strength (RSS) values at receivers for localization in an RDZ setting. The advantage of this approach is its low overhead — RSS values can be easily measured by any RF sensor. While localization techniques based on *angle-of-arrival* or *time-difference-of-arrival* can be more reliable and accurate than localization via RSS [5], these techniques require specialized hardware for synchronization or positioning, which can be unrealistic in a large scale RDZ.

Localization methods and their limitations. The classic approach towards localization is based on triangulation using the dynamics of wave propagation (see [5] for a survey). However, adding or changing the position of receivers requires (re-)solving an expensive inverse problem. Motivated by this, researchers have recently proposed data-driven, machine learning (ML) based approaches. Currently, the majority of existing work on outdoor localization is based entirely on synthetic data, or on very small datasets, both in terms of the number of transmitter locations and the geographic area

covered, typically a small area of less than 100×100 m.

Apart from requiring a large amount of training data, ML methods are also notorious for their inability to handle "out of distribution" (OOD) data, i.e., data dissimilar to those seen during training [6], [7]. In applications of localization, it is common to have significant differences between training and test, due to changes in weather, differences in transmitter and receiver types, varying channels or power levels, etc., and thus being robust to OOD data is an important requirement. Further, most known ML methods require carefully chosen model architectures and manual parameter tuning in order to achieve optimal results, leading to inductive biases and possibly overfitting to the dataset. To the best of our knowledge, there is no systematic study on understanding the robustness of localization models and the significance of parameter tuning.

Goals and contributions. Succinctly, the goal of our work is to provide a large outdoor dataset for localization research that can help evaluate the quality and robustness of data driven techniques. We wish to identify ML architectures that can work with varying numbers of uncalibrated receivers, and can also perform well on natural OOD examples.

To this end, we collect a localization dataset using the open-access POWDER testbed at the University of Utah [8], an open-access wireless testbed [8] capable of over-the-air transmission and reception with heterogeneous sensors. We study deep learning-based techniques using RSS measurements to localize transmitters. We compare different convolutional neural network (CNN) architectures from recent works [9]–[11], and study the effects of architecture and parameter choices. More specifically, our contributions in this paper are as follows:

- We present a new dataset for evaluating localization methods, covering over 4 km^2 with over 4,500 unique transmitter locations with RSS measurements from heterogeneous sensors. We also provide *splits* or partitions of the data that allow us to quantify the robustness of models.
- Using the OOD splits from our dataset, we evaluate the robustness of several localization techniques to changes in seasons, sensors, and mobility. We find that the accuracy on these samples is significantly worse than in-distribution samples, illustrating the brittleness of current techniques to "natural" changes.
- We develop a learning-based pseudo-calibration technique for inputs from heterogeneous sensors. Our learned calibration provides up to 18% improvement in accuracy for our approach, and adding this as a pre-processing step for other learning-based localization techniques such as LLOCUS [12] improves results by up to 35%.
- We propose ***CUTL***, **C**alibrated **U**-Net **T**ransmitter **L**ocalization, a U-Net-based ensemble model which uses our learned pseudo-calibration. CUTL is **49%** more accurate than other localization techniques on in-distribution data and has up to a **10%** improvement on the OOD splits in our dataset. In developing CUTL, we study the effects of network parameters and architecture.

II. Background and Related Work

In this work we only consider RSS-based localization due to the ease of collecting measurements on any device. Additionally, RSS values do not require sensors to record signals, alleviating a major privacy and storage concern. RSS-based localization has been investigated for both single transmitter [13] and multi-transmitter scenarios [5], [14]. Traditional methods rely solely on calibrated RSS measurements and an assumed RF propagation model. This makes them error-prone when the propagation model cannot capture the complex propagation characteristics of a real-world environment. To alleviate this issue, recent localization techniques [10]–[12], [15] use fingerprinting to train a machine learning model based on data taken in a specific environment.

Deep Learning for Localization: Recent promising approaches for localization utilize deep learning techniques for image processing. In general, there are two approaches for deep learning-based localization using RSS values:

1) Directly predict transmitter coordinates, as in [15].
2) Produce a 2D map of probable transmitter locations, as in [9]–[11].

These techniques have both been shown to be effective in single or multi-transmitter scenarios [9], [10], [15]. However, all of these techniques are evaluated on data either simulated using propagation models [10], [15], using ray tracing in simple virtual environments [11], or from small-scale datasets of less than 100×100 m [9], [10], [12], [14]. To our knowledge the only existing city-scale localization datasets [16], [17] use homogeneous receivers, contain few receivers within the area of interest, and only report limited sets of RSS values.

In this work, we consider a crowdsourced localization problem with heterogeneous sensors, and assume there is no available calibration data. In order to locate rogue transmitters in an RDZ, we cannot rely on techniques or data which depend on specific infrastructure or technologies, such as WiFi localization.

III. Problem Formulation

We consider localization within a geographic area which operates as an experimental RDZ. Users within this zone rely on a central *spectrum authority* (SA) to guarantee users protection from interference. If an offending transmitter must be located, sensors take energy measurements and share these measurements with the SA for localization.

For n receivers monitoring the spectrum, each measurement s_k received by the SA consists of an RSS measurement and the receiver coordinates: $s_k = [r_k, x_k, y_k], 1 \leq k \leq n$. This set of measurements $S = [s_1, s_2, \ldots, s_n]$ is the input to the localization algorithm. We assume that every measurement s_k may contain error in both the RSS measurement and the coordinates. Let the set of active transmitter locations be Q. The objective of a localization algorithm is to learn some function L that approximates Q, denoted $L(S) = \hat{Q}$.

In our setting, we assume that there is only one active transmitter per sample ($|Q| = 1$), except for a special test

TABLE I
DEVICE SPECIFICATIONS FOR TESTBED SENSORS

Category	SDR	Antenna	Count
Mobile	B210	Taoglas MA244.LBIC.002	8
Rooftop	X310	CommScope VVSSP-360S-F	6
Dense	B210	CommScope VVSSP-360S-F	5
Fixed	B210	Taoglas GSA.8841	10

case discussed later. Although other works [9], [10], [15] focus on the multiple transmitter setting, the single transmitter case remains a difficult problem, particularly for OOD data.

IV. Data Measurement

A large contribution of this work is our large-scale measurement campaign of a mobile transmitter. We recorded over 4,500 unique transmitter locations in a 2×2 km area with between 9 and 25 simultaneous receivers collecting RSS values. These measurements were taken using 29 unique sensors, with four different device and antenna configurations. In this section we describe our method of data collection, hardware details, and unique train-test separations and special test-cases that we use for evaluating localization accuracy.

A. Data Collection

In order to capture a large localization dataset, we utilized the POWDER platform at the University of Utah, an open-access wireless testbed with software defined radios (SDR) distributed across campus. We use a tool from [18] to take simultaneous RSS measurements at available receivers in the testbed.

We collect measurements in the Family Radio Service/General Mobile Radio Service (FRS/GMRS) band, which allows for 2-way voice communications. Though not used for broadband communications in most regions, the FRS band is of considerable interest due to its proximity to LoRa frequencies in Asia (433 MHz) and TV white-space and mobile broadband bands in the US and UK (ranging from 400-800 MHz). Obviously, experimental transmissions are prohibited in most bands, and unlicensed ISM bands contain far too much interference to accurately measure signal strength.

A BaoFeng BF-F8HP portable FM radio was used to transmit a narrowband audio signal in the FRS band at 462.7 MHz while walking, cycling, and riding in a vehicle with speed of less than 13 m/s (30 mph). GPS measurements of the transmitter location were taken once per second using Motorola G5 Plus and Google Pixel 2XL smartphones, which typically have error less than 10 m, but error can occasionally exceed 40 meters [19].

1) Radio Details: Each of the testbed devices belongs to one of four categories, based on the device placement and hardware specifications. These categories are *Mobile* sensors mounted in shuttles, *Rooftop* sensors on campus buildings, *Dense* sensors on 8 m tall street-poles, and *Fixed* sensors mounted on the side of buildings at ground level. Hardware details are shown in Table I.

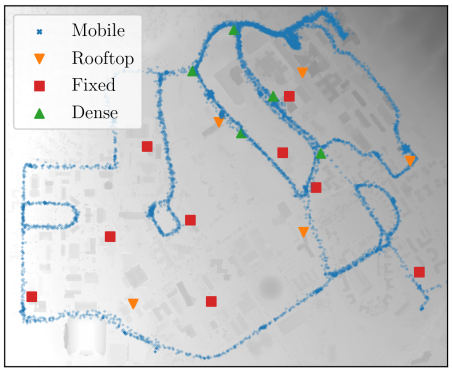

Fig. 2. A map of radio sensors in the testbed that were used for collecting signal strength measurements.

A map of receivers, including mobile shuttle routes, is shown in Fig. 2. This map illustrates that our dataset has an extremely low sensor density (at most 4 *Mobile* sensors were active at a single time). There were between 9 and 24 active sensors in any given sample, giving a sensor density between 2.25-4.0 sensors/km^2. For comparison, the sensor densities used in [10], [15] are between 100-400 sensors/km^2 in simulated data and 18 sensors in the small 0.001 km^2 testbed in [10]. The extremely low sensor density in our dataset provides a significantly more difficult localization problem than in other works.

The dataset was collected over the course of four days, with the first day in late April, and other measurements taken in early July. Each measurement consists of 10,000 IQ samples recorded at a sample rate of 2 MHz. After collection, a 6 kHz bandpass filter was applied to the samples. As is standard, the signal power was calculated as the average sum of absolute squares. The gain of each receiver was set to 35 dB, though this can represent either a medium or high gain depending on the device. For example, the B210 radios had a maximum gain of 76 dB, and the X310 radios had a maximum RX gain of 37.5 dB. Without calibration between these radios, it is unknown how the gain settings of different sensors relate.

Unlike other localization datasets [10], [12], [13], [16], our dataset consists of measurements from heterogeneous devices. The radios used are high-end Ettus SDRs, but variation in gain settings, antenna placement, and antenna radiation pattern all provide a varying set of sensors which makes localization an even more challenging problem.

B. Train-Test Separations

To our knowledge, this is the largest scale localization dataset with heterogeneous sensors existing in the literature. In order to fully explore data captured in this complex environment, we create several divisions of our dataset intended to assess the robustness of localization techniques.

The first and most obvious split is a random 80/20% separation of a train and test set, which we refer to as the *Random* split. The other splits of our dataset are meant to

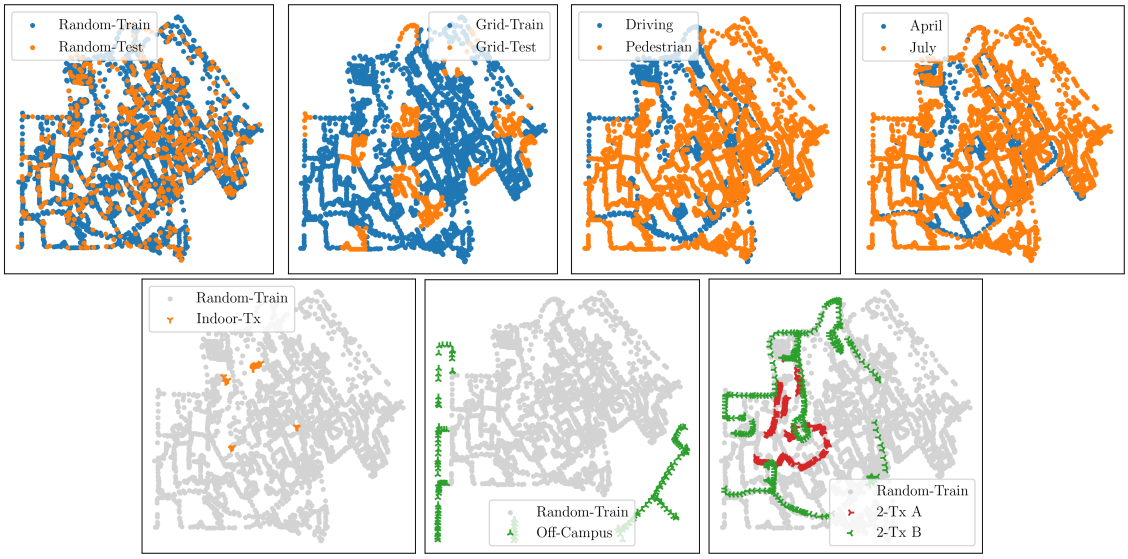

Fig. 3. Maps of transmitter locations for each dataset split. L-R, top-bottom: *Random*, *Grid*, *Driving*, *Seasonal*, *Indoor*, *Off Campus*, and *2-Tx*.

approximate realistic deployment challenges. These are shown in the top row of Fig. 3.

- *Grid:* We divide campus into a 10×10 grid, where 20 rectangles are randomly assigned to the test set, and the remaining 80 are assigned to the training set. The *Grid* split is used to demonstrate the performance of localization techniques in unseen regions.
- *Driving/Pedestrian:* The transmitters used in this dataset were carried either while walking, cycling, or riding in a car. The *Pedestrian* set includes cylcing.
- *April/July:* We separate this data based on the date of collection, with one day in April and three in July. We use this data split to observe generalization over time.

We also consider three special cases which are meant to evaluate localization techniques on data that is significantly different from the training distribution, shown in the bottom row of Fig. 3. Similar to how computer vision researchers introduced ImageNet-C [6] to measure a model's robustness to OOD data, we evaluate models trained on the *Random* data split on these special cases to measure model robustness.

- *Indoor:* A small set of samples with an indoor transmitter near windows.
- *Off-campus:* A small set of samples were taken far outside the campus boundaries far from most receivers.
- *2-Tx:* A small set of samples taken with two active transmitters at a time.

We make this dataset and the train/test splits outlined here publicly available [20]. We hope that this data will become standard for evaluating future localization techniques on uncalibrated, heterogeneous sensors, since our results show that current localization techniques are generally not robust to OOD data.

V. Methodology

We use deep learning to avoid making any assumptions about the environment or transmitter characteristics, instead learning directly from sensor data without interpolation or estimation, at the risk of biasing a model based on our data. We test variants of the U-Net architecture [21] for localization, In the rest of this section, we detail our localization technique, including network architectures, sensor calibration, and the training and evaluation process.

A. Localization via Image Transformation

We view localization as a computer vision task, where the set of receiver measurements S is represented as an image, and the objective is to either (1) directly predict the transmitter coordinates, or (2) generate a corresponding image that shows the position of active transmitters. This image setting captures the spatial relationship between receivers. To form an input image X, the pixel corresponding to each sensor measurement $s_k \in S$ is set to the normalized RSS value, and all pixels without a sensor value are set to 0. We follow a similar process to form a the target image Y. All pixels are set to 0 except for a 3×3 square with a center value of 1 and exterior values of 0.5. This image-to-image localization process is shown on the left side of Fig. 4, where the measurements and locations are converted into input and target images X and Y.

This process requires discretizing the coordinate space, in terms of meters per pixel. As the meters per pixel decreases, the processing time increases quadratically, and accuracy can suffer if the network architecture is not designed for a large input image. On the other hand, if the meters per pixel is high, then the loss of precision can harm localization accuracy. An exploration of results with varying meters per pixel is presented in our results.

1) Architecture and Optimization : Our U-Net model consists of a series of downsample and upsample blocks, which

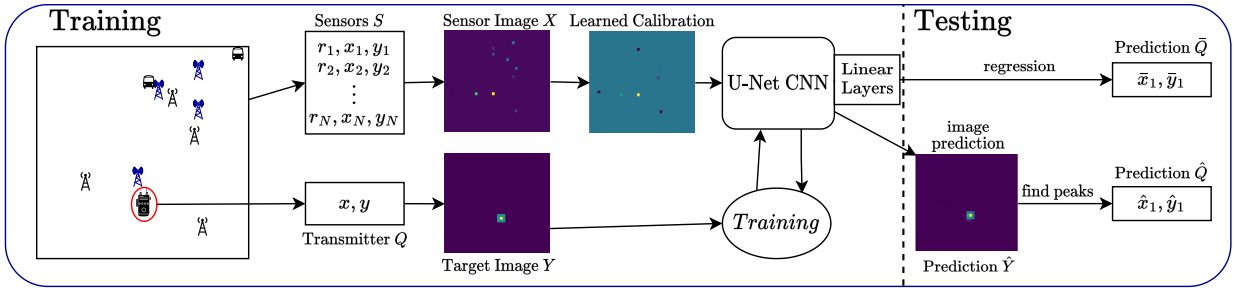

Fig. 4. The CNN image training pipeline. Sensor and transmitter data S and Q are made into the input and target images X and Y, which are used to train the CNN. The CNN is trained to output either a direct prediction $\bar{Q}$, or an image prediction $\hat{Q}$.

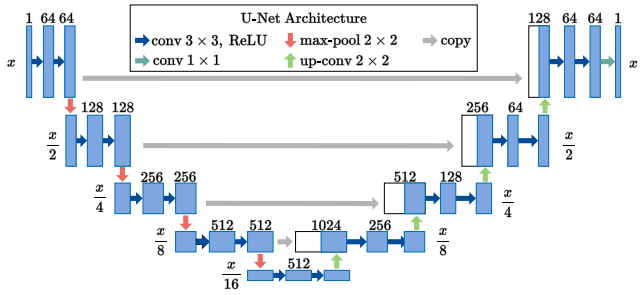

Fig. 5. Our variant of the U-Net architecture

use convolutional layers to encode feature channels, along with residual connections between layers to preserve the high-resolution information from early layers. This network architecture is shown in Fig. 5.

We explored several variants of U-Net in our study, including deeper and shallower networks, larger convolutional kernels, no residual connections, and a downsampling-only version with additional linear layers for output. All these models performed similar to or worse than our chosen architecture.

We consider both direct coordinate prediction where the error in the (x, y) coordinate predictions is minimized, and image-to-image prediction where the difference between the target image Y and predicted output $\hat{Y}$ is minimized. As will be shown in our results, both techniques are effective at localization, which we find somewhat surprising, since in the image-to-image setting a prediction with 5 pixels of error receives the same penalty as a 50 pixel error. In spite of this, our most accurate model is trained using this technique.

It may be noted that the Wasserstein or earth-mover's distance (EMD) is an ideal loss function for the image-to-image prediction, since it would penalize a 50 pixel error more. However, EMD is a non-trivial objective to compute (requiring solving a matching problem), and to our knowledge no efficient implementations of EMD-approximations exist for 2-dimensional distributions.

2) Training and Evaluation: We use the training and test splits outlined in Section IV-B, randomly selecting a subset of the training samples as validation data. This validation set is used to select model hyperparameters such as pixel scale

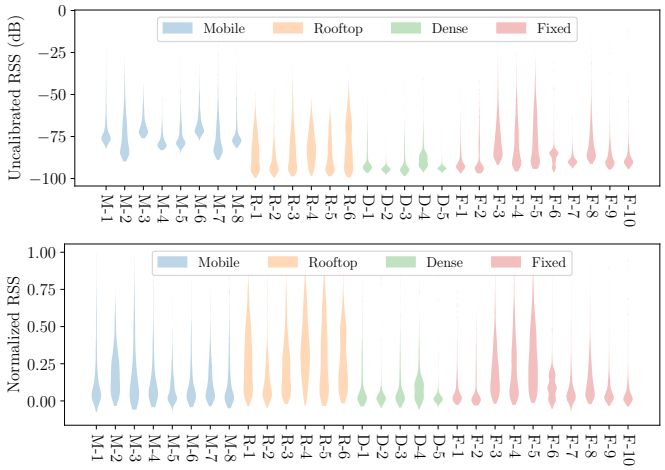

Fig. 6. Violin plot of the RSS measurements, without normalization (top), and with sensor-specific normalization (bottom). Each vertical shape represents the distribution of the RSS values for each sensor.

and training epochs. We train five different models for cross-validation with random validation sets, then train and evaluate a model using the full training set and the chosen parameters.

We also consider a *bagged* ensemble of the five cross-validation models models, as in Breiman's classic work [22], where we calculate a weighted average of predictions from each of the five models. This weighted average enables a more precise sub-pixel prediction.

B. Signal Strength Calibration

Without calibration data available, we normalize and learn to calibrate our sensors. In Fig. 6, we show the distribution of RSS values for the different sensors in our dataset, with the sensor category shown by the color. The RSS values before normalization are shown in the top plot. Note that the *Mobile* sensors on the left have wide variation in the RSS distributions, particularly in the noise-floor.

We normalize the RSS values before they are input to the model. Each sensor has a unique noise floor, so the RSS value is normalized on a per-sensor basis based on the measured noise floor and the maximum observed RSS value. After normalization, Fig. 6 (bottom) shows that there is still large variation in the RSS distributions, due to environmental

diferences such as line-of-sight, obstacles such as buildings, and low coverage of the edges of campus.

In order to compensate for the lack of calibration between different radios, we scale the RSS values in order to improve localization performance. Simply, we apply a weight w_i and bias b_i to scale the RSS value from each sensor,

$$\bar{r_i} = w_i r_i + b_i$$

We use gradient descent to learn these two values during the localization training process, with unique parameters either for each *sensor*, or for each of the four *categories* of sensors. Adding a constant bias b_i is the standard method of calibrating device power in dB [23]. we also include a multiplicative weight to scale the importance of each sensor.

We found that in certain cases calibration made modest improvements to the accuracy of our model as well as other localization techniques. The impact of this calibration is discussed fully in Section VI.

VI. Evaluation and Results

In this section we evaluate localization methods on our data. We train and evaluate various CNN architectures. We consider different pixel scales and pseudo-calibration techniques and show their impact on accuracy. We also explore how models perform on OOD data by evaluating CNNs on the train/test splits outlined in Section IV-B.

In our evaluation, we reference the median error rather than mean when evaluating localization techniques. In our results, the mean is 1.2-1.65× greater than the median. We find that median provides a more clear estimate of typical error.

A. Architecture Evaluation

In Fig. 7, we compare the median error of different network architectures on the *Random* split of our data. We compare the following models:

- *U-Net*: The 19-layer downsample-upsample CNN shown in Section V, using the image-to-image loss function.
- *U-Net+Linear*: The same U-Net architecture with three linear layers for direct coordinate prediction.
- *DeepTxFinder* [15]: A direct coordinate prediction technique with four convolutional layers and one linear layer.
- *DeepMTL* [10]: An image-to-image prediction technique using four convolutional layers. The authors use a highly complex sub-pixel prediction technique using object detection that was not implemented for this work.

1) Pixel Scale: More than any other parameter, the meters-per-pixel drastically affects localization accuracy. We consider pixel scales between 5 and 140 m per pixel, shown in Fig. 7. The U-Net model outperforms all others at 30 m per pixel, with the most accurate pixel scales for DeepTxFinder and DeepMTL at 100–110 m. The U-Net+Linear performs well with less dependence on a specific pixel scale.

Fig. 7 only shows the results for the *Random* split of data. In the other splits, discussed later, U-Net performs better with a larger pixel scale of 60 to 80 m. Because of the better performance on OOD data and in order to provide a simpler practical model, we use a constant pixel scale for each model: 60 m for U-Net and U-Net+Linear, 100 m for DeepMTL, and 110 m for DeepTxFinder.

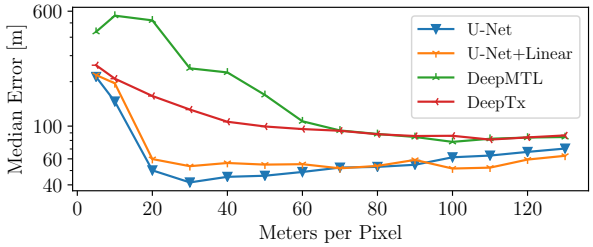

Fig. 7. Comparison of CNNs with varying pixel scale.

Fig. 8. U-Net prediction with pixel scale of 10 m (left) and 60 m (right).

The poor performance at low meters-per-pixel was surprising. This seems to be due to the difficulty of training a loss function at high resolution. One example output of models trained at 10 m and 60 m pixel scale is shown in Fig. 8. With predicted values between 0 and 1, we interpret each pixel value as a "confidence" that a transmitter exists in that location. The 10-m prediction (left) has a max confidence of 0.03 and an error of 154 m, compared with the 60-m prediction on the right, with a confidence of 0.82 and an error of 17 m. The 10-m prediction has several clusters of relatively high-valued predictions which are centered around sensors, but the 60-m prediction has only one confident prediction.

We conjecture that poor performance at low pixel scale is due to the limited depth of CNNs. The number of pixels between transmitters and receivers is significantly larger at high resolution, making it difficult to learn a function representing the relationship between transmitter and receiver.

2) Bagging : In Fig. 9, we compare the performance of the bagged models (5-ensemble) with models trained on all of the training data. The importance of each of the five predictions is scaled by the model confidence for the prediction, rather than just taking the average of the five predictions. This weighted average is a type of sub-pixel prediction, so it provides a greater improvement in mean error for the image-to-image predictions (U-Net and DeepMTL) compared to the direct coordinate prediction (U-Net+Linear and DeepTxFinder).

3) Model Confidence and Error : An important question for evaluating a machine learning model is identifying cases where the model fails to perform well. We expected that the highest localization error would be for transmitters that were far from the nearest receiver. However, Fig. 10 shows only a weak correlation between the distance between a transmitter and the nearest receiver and the localization error ($R = 0.16$). Some outliers have been removed to show the scale more

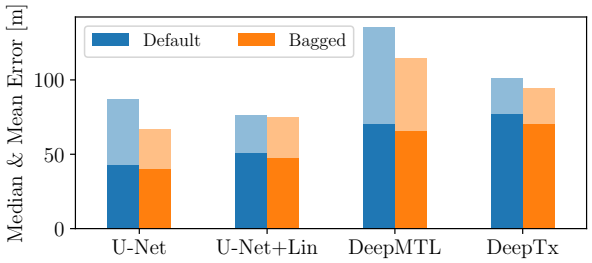

Fig. 9. A comparison of Full or Bagged models, with mean error shown in light color.

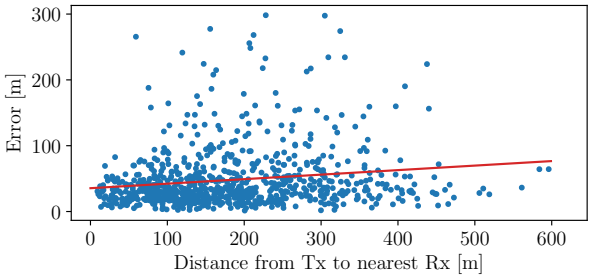

Fig. 10. Comparison of distance between Tx and Rx against localization error.

clearly. Overall, this result is very surprising: the distance between a transmitter and receiver does not have a strong effect on the localization error. We conjecture that these high-distance points may have a unique fingerprint that is more easily identified by our model.

B. Out-of-Distribution Performance

In order to evaluate how CNN-based localization generalizes to cases that are outside the distribution of training data, we train separate models on the train/test splits described in Section IV-B. We report full results for the boosted U-Net model at 60 m per pixel in Table II, including training and test set details and the median error on each test set. In the rightmost two columns, we also report the prediction error and the percentage of the test set which have confidence over 0.5.

TABLE II
THE MEDIAN LOCALIZATION ERROR FOR THE U-NET MODEL FOR ALL TEST SETS DESCRIBED IN SECTION IV-B. THE RIGHTMOST TWO COLUMNS CONTAIN RESULTS WHERE ONLY PREDICTIONS WITH CONFIDENCE ABOVE 0.5 ARE CONSIDERED.

	Train	Train	All Samples		Conf. ≥ 0.5	
Test Set	Set	Size	Err. [m]	Size	Err. [m]	% of Test
Rand.	Rand.	3399	40.1	828	39.2	98%
Grid	Grid	3536	117.6	691	111.9	92%
Pedest.	Drive	925	181.5	3302	142.4	53%
Drive	Pedest.	3391	264.9	836	244.4	83%
July	April	811	207.4	3416	164.5	27%
April	July	3416	335.8	811	255.0	45%
2-Tx 1st	Rand.	3399	160.6	346	148.4	89%
2-Tx 2nd	Rand.	3399	466.4	346	457.1	3%
Indoor	Rand.	3399	126.3	89	91.5	90%
OffCamp	Rand.	3399	566.1	156	439.0	63%

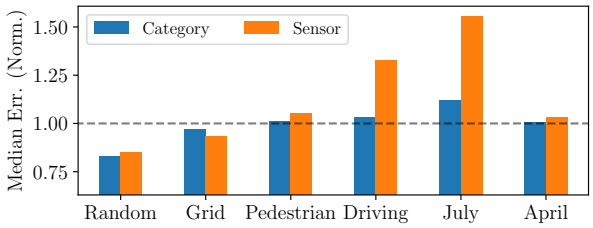

Fig. 11. Effect of learned calibration applied to each Category or each unique Sensor, normalized by error with no calibration.

1) Train-Test Splits : As expected, localization on data from outside the training distribution results in poor accuracy. Of these OOD cases, the most accurate model was trained on the *Grid* split, which has a median error of nearly 3× the *Random* split error. In the worst case, the *Off-Campus* test set has an error of 566.1 m, over 14× the *Random* split error.

Table II shows that a larger training set does not indicate better performance. Models evaluated on the *Pedestrian* and *July* test sets were each trained on less than 1000 training samples and evaluated on over 3000 samples, yet they both have a median error 80-130 m less than when training and test sets are reversed. Increasing the size of the training set seems to help accuracy only if the distribution of test data is similar. Otherwise overfitting is likely.

One interesting case is the *2-Tx* test set, which had two active transmitters. For the first prediction (i.e., the maximum peak in the array), the median error is 160.6 m, lower than many of the OOD data splits. The second prediction had a median error of 466.4 m, with only 3% of these predictions having a confidence over 0.5. This indicates that with two transmitters active, a model trained on single-transmitter data may still perform well for finding one transmitter.

2) Signal Strength Calibration : Since our sensors provide uncalibrated RSS values, we can apply a pseudo-calibration technique using learned calibration parameters. We consider two different calibration techniques: learn linear parameters for each **category** of sensors, or learn linear parameters for each individual **sensor**. In Fig. 11, we show the effectiveness of these techniques on each of our six dataset splits using the bagged U-Net model, normalized by the error with no calibration applied.

We see a notable improvement of up to 18% in the *Random* split using category-specific calibration, and minor improvement in the *Grid* split, but no improvement using the other techniques. This illustrates an important point: for data that is significantly different than the training distribution, allowing the model to learn an extra parameter for calibration actually just enables overfitting. Sensor-specific calibration allows an even greater degree of overfitting.

The learned calibration parameters provide an interesting insight into how valuable a model finds data from different sensors. Fig. 12 shows the RSS distributions of each sensor after calibration. The most obvious effect of calibration is shifting the *Mobile* RSS values downward by 0.26, decreasing

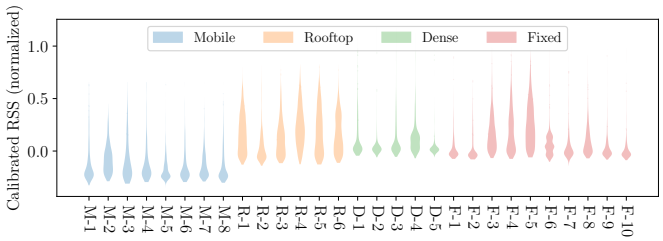

Fig. 12. Device RSS distributions after normalization and category-specific calibration.

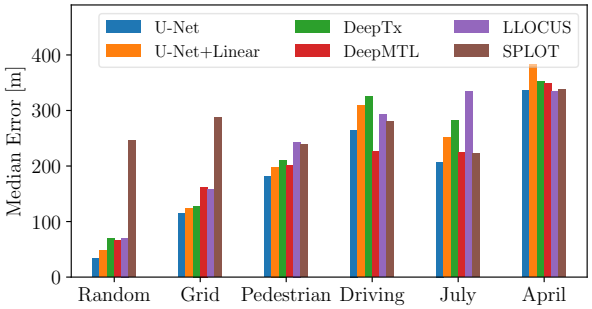

Fig. 13. Comparison of localization techniques across all dataset splits.

their impact. In our observations, the *Mobile* sensors were often noisy, potentially due to moving closer to interference sources as well as electrical noise on the shuttles.

3) Non-CNN Localization: We compare against two RSS-based localization techniques which do not use deep learning:

- *LLOCUS* [12]: Interpolate sensor data to a set of fixed locations, then train a simple ML model (k-nearest-neighbors or radial basis function) forr localization.
- *SPLOT* [14]: Use matrix-inversion with a path-loss model to estimate the transmit power field over the space, where the maximum field value is the predicted location.

The localization algorithms used in LLOCUS and SPLOT both explicitly require that input values are a calibrated power measurement, which may not be available. In such a case, our learned pseudo-calibration can improve accuracy. We applied our learned-calibration function to LLOCUS and SPLOT. Using category-specific calibration parameters from a trained U-Net model improved the median accuracy for LLOCUS between 10-35%. Unlike in Fig. 11, calibration improved LLOCUS accuracy for all dataset splits. Results for SPLOT are much more modest, improving by only 1% on average.

4) Results on All Splits: Fig. 13 shows the results of all tested localization techniques on the six train-test splits, with ideal pixel scale applied for the deep-learning models, and sensor-category psuedo-calibration applied to LLOCUS. U-Net outperforms previous techniques by 49% on the *Random* split, as well as better accuracy on the majority of dataset splits, with one exception of the *Driving* test set where DeepMTL achieves a 39 m advantage.

No technique achieves sub-100 m accuracy (roughly 5% of the width of campus) in any of the OOD test splits, but we note

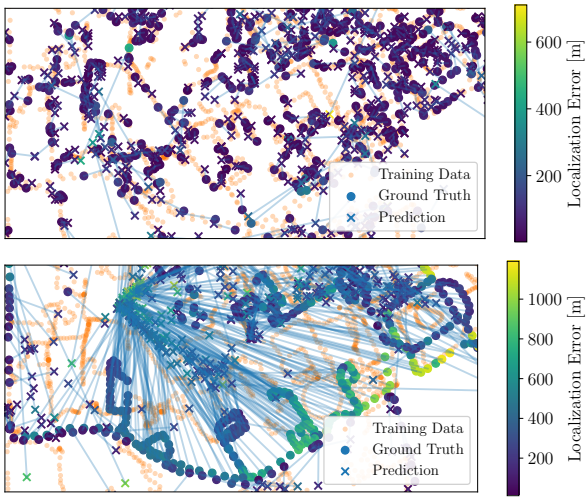

Fig. 14. Plots of predictions, ground truth, and training data for the *Random* (left) and *Driving* (right) sets. *Error-vector* lines connect ground truth and predicted locations.

that this is an expected outcome. Due to the complexity of the campus environment, noise in RSS and GPS measurements, the difficulty of learning to perform on unseen distributions of data, and the extremely low sensor density, we conjecture that no existing localization technique can perform significantly better in these OOD circumstances without either additional information about the environment or other mechanisms to correct bias.

5) Detailed Error Analysis : Statistics such as the mean and median offer a good understanding of the typical behavior of a localization technique, but they do not provide a complete picture. In Fig. 14, we show maps illustrating prediction error for the *Random* and *Driving* test sets using the bagged U-Net model at 60 m per pixel. Faint orange dots represent locations in the training set. Locations in the test set are shown by larger circles, where the circle color indicates the error of our model predictions. Each model prediction is shown by a colored 'X', with a thin line drawn between each prediction and the ground truth. This *error vector* line can be helpful in illustrating any trends in our prediction error.

For the *Random* data (top), the predictions are generally very accurate, with just a few notable predictions with high error. On the other hand the *Driving* test set has much higher error; the error also has a clear structure. By looking at the blue error lines between ground truth and predictions, we can observe there are some clear clusters of predictions in the image. These clusters indicate a bias in the model predictions. This may be due to bias in the training data, which was taken on foot, allowing for a much greater density of training points in certain locations.

C. Discussion

One of the greatest challenges in this localization setting is the extremely low sensor density, orders of magnitude lower than other localization techniques were designed for. In

light of this, it becomes apparent that evaluating localization techniques on simulated data with impossibly high sensor densities is not representative of the expected performance of these techniques in real-world scenarios. This is why over-the-air experimental datasets such as the one provided with this paper are crucial to evaluating transmitter localization techniques.

Our proposed U-Net model outperforms all other evaluated techniques, though all models have significantly lower accuracy on OOD data compared to the *Random* split. Generalizing to completely unseen data is the paramount challenge of machine learning, and our deeper network provides a significantly more accurate model in both the in-distribution *Random* cases as well as most of the OOD splits. When we restrict predictions to those with high confidence (> 0.5), we have even more accurate predictions. Our results also highlight the importance of choosing an appropriate pixel scale.

Our learned pseudo-calibration is a simple, effective method that allows for heterogeneous sensors to be used for localization, and can even be applied to other techniques which require calibration. The ability to learn parameters that calibrate an entire category of sensors is extremely useful. If parameters have been learned for an existing sensor category, then adding a new sensor of the same type becomes trivial.

As a result of these experiments, we propose CUTL, or Calibrated U-Net Transmitter Localization as a new localization technique. CUTL uses learned psuedo-calibration to scale RSS inputs and predicts locations using a bagged ensemble of U-Net models.

VII. Conclusion

With very few large outdoor datasets for localization available, we contribute a large dataset for localization containing over 4,500 unique transmitter locations and heterogeneous sensors. We make our dataset publicly available for research and evaluation, including non-random splits of our dataset used to evaluate localization performance on OOD data. Without evaluation on such challenging cases, localization techniques cannot be shown to be robust to realistic changes in seasons, sensors, or areas of interest.

We have used CNNs to localize a single transmitter in our outdoor dataset using RSS measurements, including comparison of direct coordinate prediction and image-to-image models for localization. Our CUTL technique outperforms previous methods through the use of a deeper network, ensemble models, and pseudo-calibration which scales inputs from heterogeneous devices. Our future work will focus on using context about the environment to further improve accuracy and robustness.

References

[1] D. Zhu, J. Li, and G. Li, "Rfi source localization in microwave interferometric radiometry: A sparse signal reconstruction perspective," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 6, pp. 4006–4017, 2020.

[2] A. G. Dempster and E. Cetin, "Interference localization for satellite navigation systems," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1318–1326, 2016.

[3] N. S. Foundation, "Spectrum innovation initiative: National radio dynamic zones," March 2022.

[4] S. J. Maeng, İ. Güvenç, M. L. Sichitiu, and O. Ozdemir, "Out-of-zone signal leakage sensing in radio dynamic zones," in *ICC 2022-IEEE International Conference on Communications*, pp. 5579–5584, IEEE, 2022.

[5] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[6] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *Proceedings of the International Conference on Learning Representations*, 2019.

[7] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet?," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400, PMLR, 09–15 Jun 2019.

[8] J. Breen, A. Buffmire, J. Duerig, K. Dutt, E. Eide, A. Ghosh, M. Hibler, D. Johnson, S. K. Kasera, E. Lewis, *et al.*, "Powder: Platform for open wireless data-driven experimental research," *Computer Networks*, vol. 197, p. 108281, 2021.

[9] F. Mitchell, A. Baset, N. Patwari, S. K. Kasera, and A. Bhaskara, "Deep learning-based localization in limited data regimes," in *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, pp. 15–20, 2022.

[10] C. Zhan, M. Ghaderibaneh, P. Sahu, and H. Gupta, "Deepmtl pro: Deep learning based multiple transmitter localization and power estimation," *Pervasive and Mobile Computing*, 2022.

[11] Ç. Yapar, R. Levie, G. Kutyniok, and G. Caire, "Locunet: Fast urban positioning using radio maps and deep learning," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4063–4067, IEEE, 2022.

[12] S. Sarkar, A. Baset, H. Singh, P. Smith, N. Patwari, S. Kasera, K. Derr, and S. Ramirez, "Llocus: learning-based localization using crowdsourcing," in *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pp. 201–210, 2020.

[13] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on signal processing*, vol. 51, no. 8, pp. 2137–2148, 2003.

[14] M. Khaledi, M. Khaledi, S. Sarkar, S. Kasera, N. Patwari, K. Derr, and S. Ramirez, "Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pp. 235–247, 2017.

[15] A. Zubow, S. Bayhan, P. Gawłowicz, and F. Dressler, "Deeptxfinder: Multiple transmitter localization by deep learning in crowdsourced spectrum sensing," in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–8, IEEE, 2020.

[16] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, "Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas," *Data*, vol. 3, no. 2, p. 13, 2018.

[17] E. Alimpertis and A. Bletsas, "CRAWDAD dataset tuc/mysignals (v. 2019-10-30)." Downloaded from https://crawdad.org/tuc/mysignals/20191030, Oct. 2019.

[18] K. Webb, S. K. Kasera, N. Patwari, and J. Van der Merwe, "Wimatch: Wireless resource matchmaking," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, IEEE, 2021.

[19] T. Szot, C. Specht, M. Specht, and P. S. Dabrowski, "Comparative analysis of positioning accuracy of samsung galaxy smartphones in stationary measurements," *PloS one*, vol. 14, no. 4, p. e0215562, 2019.

[20] A. Authors, "A Dataset of Outdoor RSS Measurements for Localization," Oct. 2022.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[22] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[23] P. Horowitz and W. Hill, *The art of electronics; 3rd ed.* Cambridge: Cambridge University Press, 2015.