2024 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN) | 979-8-3503-1764-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/DySPAN60163.2024.10632736

Utilizing Confidence in Localization Predictions for Improved Spectrum Management

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

Jie Wang
McKelvey School of Engineering
Washington University in St. Louis
St. Louis, Missouri, USA

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

***Abstract*—Transmitter localization is an important component of next-gen spectrum sharing and management systems. Recently, machine learning (ML) methods have shown promising results for localization in complex environments. However, existing ML models have significant limitations, such as the need for careful parameter tuning and the lack of accuracy on out-of-distribution (OOD) examples. Moreover, current ML models do not typically provide a "confidence" in their prediction. In this work, we propose a new training procedure based on the Earth Mover's Distance (EMD) that improves on these limitations. The method improves OOD accuracy by up to 22% while providing more interpretable results. The EMD-trained model also produces a confidence score, which can be used to identify high-error and OOD examples. We demonstrate how model confidence serves as a guide for *hybrid* localization models. This includes selecting the most reliable prediction from multiple models based on confidence values or resorting to a fallback path loss technique in cases of low confidence. Our work establishes the importance of model confidence in improving the accuracy of localization and as a mechanism for effective decision making in localization applications.**

***Index Terms*—transmitter localization, uncertainty, model robustness, spectrum management**

I. Introduction

Radio Dynamic Zones (RDZ) are envisioned as a platform for next-gen spectrum sharing [1]–[3]. They facilitate secondary spectrum usage inside the zone while ensuring interference protection outside the RDZ for primary/incumbent users, including those sensitive to radio-frequency (RF) interference. To aid in managing spectrum in the RDZ, a digital spectrum twin (DST) has been proposed [4], [5] as a system that uses environmental features and measurements of current RF conditions to simulate user activity in the RDZ. Fundamentally, a DST serves as a platform for simulating an RF environment based on current, real-world conditions. This may involve a sophisticated propagation model that uses physics-based or ML techniques to estimate signal strength based on environmental factors such as building elevation and ground surface type [5]. These environmental simulations allow RDZ management systems to make informed decisions on operations in the RDZ, such as determining the location and power level at which a user is allowed to transmit in order to protect sensitive users both within and outside the zone. To ensure the protection of sensitive users, a fundamental problem is to localize transmitters, especially ones that may be causing interference [2] by utilizing real-time data from sensors in the area of interest, as shown in Fig. 1.

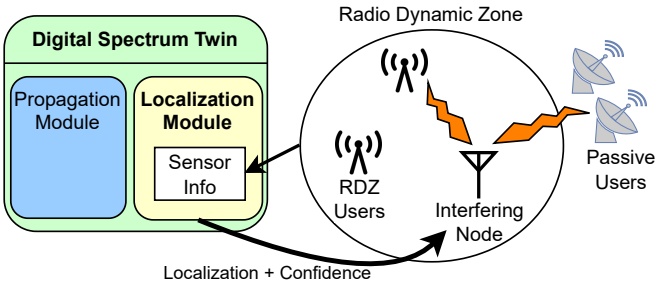

Fig. 1. A diagram of the Radio Dynamic Zone, with a Digital Spectrum Twin managing spectrum and locating a transmitter.

While ML techniques provide a promising approach for accurate localization in complex environments, previous work has revealed that ML models can be biased by their training data and fail when there is a distribution shift between the training data and input samples encountered during deployment [6]–[9]. This shift can be attributed to changes in the environment, alterations to the sensors used for localization, or the presence of transmitters in regions not covered by the training data. Deploying an ML localization model on these out-of-distribution (OOD) input samples is likely to produce uncertainty in predictions, as these inputs differ significantly from the data observed in the training set. In such scenarios, providing RDZ management systems with context about prediction uncertainty can help to allocate resources properly, inform enforcement actions, and enable hybrid localization techniques. *Our work aims to develop localization techniques that provide this vital context about prediction confidence to a spectrum management system and make more accurate predictions.*

Despite tendencies toward model bias, deep learning models remain the most effective techniques for localization [7], [10]. In this work, we find that existing ML methods for localization

This work was supported by the National Science Foundation as part of awards CNS-1827940 and CNS-1564287, as well as the PAWR Project Office grant 10046930.

have the following fundamental problems:

1) Localization models using regression are typically accurate but fail to provide interpretable results [9], [11].
2) Models that provide more interpretable results are more difficult to train since they solve an image approximation problem instead of directly minimizing the localization error during training [7], [10].
3) No existing localization techniques are intended to inform users about model uncertainty or the reliability of predictions. Understanding when a prediction is reliable is crucial for effective spectrum management.

We make progress on all of these concerns by using Earth Mover's Distance (EMD) [12] as a loss function to train localization models (see Section III for definitions). EMD captures the geometry of the localization setting by directly minimizing localization error while providing intuitive, interpretable predictions. Using EMD to train models improves accuracy on OOD test sets by up to 22% compared to other loss functions, and it also provides a score for *model confidence*.

In Section IV, we use model confidence to identify and understand localization predictions with high error. We investigate possible indicators of model confidence and then show how confidence can be used to identify high-error and OOD samples, which typically have low confidence. To understand how and why localization models have high error in some instances, we investigate specific samples from our dataset and observe that (1) ML models do not generally make predictions in locations that were not seen in the training data, and (2) low-confidence predictions often include some interference or other pathological behavior in the inputs.

Finally, in Section V we consider two practical examples of how confidence can provide more accurate localization predictions via hybrid localization models. We use a simple physics-based localization technique as a fallback to improve accuracy on OOD inputs, with minimal impact on accuracy for in-distribution samples. We also show how confidence can be used to select reliable predictions from multiple pre-trained models.

This work establishes **(1)** how EMD loss can be used to train more effective models, **(2)** how model confidence can be used to identify OOD and high-error samples, and **(3)** how model confidence enables hybrid localization models with more accurate predictions.

II. Background and Related Work

Transmitter localization in an RDZ can enable better management of the RF spectrum. Users within an RDZ may cause harmful interference to spectrum users inside or outside the zone, so employing localization to detect and identify interference sources is necessary for operating an RDZ, as noted by Maeng et al. [2]. Accurate location information allows for higher opportunistic spectrum reuse since interference can be reliably estimated based on the transmitter's position in the environment, as in [13].

This work considers localization using only received signal strength (RSS) observations. One primary advantage of RSS localization is that any low-cost RF device such as [14] can capture RSS values to achieve large-area sensor coverage at a lower cost. While we only consider RSS localization, the principles of EMD loss, model confidence, and hybrid localization, which we introduce in this work, can be applied to any localization technique and are not specific to RSS localization.

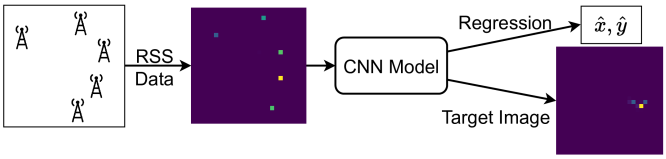

Fig. 2. The CNN localization pipeline. Sensor data is input as an image to the CNN, which estimates the target location either via regression or as an image.

A. RSS Localization

At a basic level, RSS localization can be seen as either a physics-based propagation problem, where transmitter coordinates are estimated based on physical models of propagation in the environment, or as a data-based learning problem, where ML models are trained to infer the transmitter's location based on RSS measurements without any physical models. The most straightforward data-based approach is fingerprinting, where RSS measurements are interpreted as a unique identifier of the transmitter location. Accuracy in fingerprinting and other data-based approaches depends on the assumptions that (1) nearby locations will have similar fingerprints and (2) the training data sufficiently covers the region of interest. Additionally, this reliance on training data means data-based methods may inherit bias that exists in the training data due to lack of coverage, measurement errors, or interference [15].

Most recent works in RSS-based source localization focus on data-driven approaches. Simple ML models for fingerprinting, such as k-nearest-neighbors and random forests have been shown to be more accurate than path loss-based trilateration [16]. In [17], Sarkar et al. interpolate sensor values from mobile sensors to a set of known locations, which allows for sensor mobility while using ML fingerprinting.

Current state-of-the-art techniques use significantly more complex models for data-based localization. A convolutional neural network (CNN) is used in [7], [9]–[11], [18], where RSS values are converted into a 2D image and input into a CNN model. The CNN either directly predicts the transmitter(s) coordinates [7], [9], or outputs a target image showing probable transmitter locations [10], [18]. This process is shown in Fig. 2.

In our previous work [7], we explored the robustness of CNN localization on OOD samples, where the test inputs differ significantly from the training distribution. We showed that localization accuracy is 4-10$\times$ worse on OOD samples, implying that although CNN methods are highly effective at memorizing the distribution of the training set, they fail to generalize to unseen distributions. Whether this distribution

TABLE I
DETAILS ON THE LOCALIZATION DATASET FROM [19]

Data Separation	Tx Locations	Number of Sensors
Random/Grid	4577	9 - 25
April	811	6 - 7
July	3415	17 - 25
November	351	23

shift occurs through natural environmental changes [7], bias or coverage gaps in the training data [15], or adversarial attack [8], practical ML models should be robust to some distribution shift. If this is not possible, the model should alert the user that a specific prediction may be unreliable. In this work, we improve existing CNN methods by using model uncertainty to identify unreliable predictions and improve OOD accuracy.

B. Datasets for Evaluation

In this work, we explore model localization on OOD samples from our previously published localization dataset [19], collected on the University of Utah campus in Salt Lake City, Utah. This dataset consists of GPS coordinates for a single mobile transmitter at 462.7 MHz (near several LoRa bands in Asia and TV white space in the US and UK), with GPS coordinates and uncalibrated RSS values from stationary and mobile sensors. Transmitter locations cover approximately 3 km^2.

To understand localization in both in-distribution and OOD settings, we create separations of the dataset into training and test sets, similar to those in [7].

1) **Random:** Select 20% of the data uniformly at random as the test set and the remaining 80% for training.
2) **Seasonal:** Separate data based on the collection date. Tx samples were collected in April, July, and November without mobile sensors. In most cases, we do not use data from April due to the low number of sensors.
3) **Grid$N$:** Split the region into a $N \times N$ grid. For 80% of the grid cells, assign samples with the Tx location within the cell to the training set, and do the same for the test set and the remaining 20% of cells.

Location and sensor counts for each of the data separations are listed in Table I, and plots of transmitter locations for each separation are shown in Fig. 3. The *Random* separation is the best-case scenario often used in ML research but is quite unrealistic since samples seen in deployment will never be identically distributed to those seen during training. The *Seasonal* split captures distribution shift between data captured on sunny days in July and during a winter storm in November. The *Grid2* and *Grid5* separations simulate gaps in the coverage of training data, such as might occur when samples cannot be collected in certain areas. This allows us to study OOD samples with a known and controlled distribution shift.

III. TRAINING LOCALIZATION MODELS USING EARTH MOVER'S DISTANCE

Although previous works show the effectiveness of CNN localization, results have been mixed on what loss function should be optimized during the training of the model. This section explores the loss functions used in previous works, highlighting the main problems with existing work. We propose using EMD [12], also known as the Wasserstein-1 metric, as a better-motivated loss function. We evaluate the EMD loss formulation and find that EMD is well suited for our localization setting, though other loss functions may be preferred depending on the practical application.

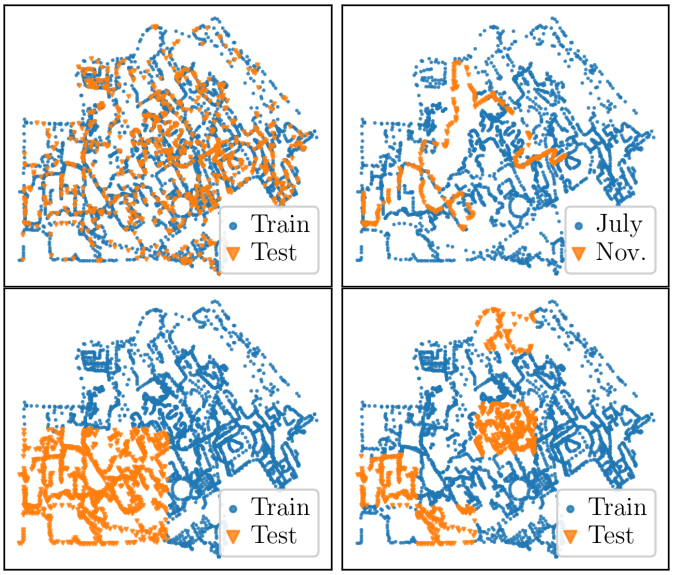

Fig. 3. Maps of Tx locations for the *Random*, *Seasonal*, *Grid2*, and *Grid5* data separations.

A. Current Training Techniques

Training a CNN model requires a loss function, or an objective to be minimized during training. Existing works with CNN localization use squared error as a loss function, though with different formulations for predicting transmitter coordinates. There are two previous techniques that have been used for CNN localization, both shown previously in Fig. 2:

1) **Regression** models [9], [11] which directly predict transmitter coordinates and learn to minimize the localization error (Euclidean distance) during training.
2) **Target image** models, which produce an image marking probable transmitter locations [10], [18]. At inference time, the highest magnitude pixel is selected as the transmitter location.

In the target image setting, the training process minimizes the following loss function:

$$L_C(\hat{Y}, Y) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} |Y_{ij} - \hat{Y_{ij}}|^2, \quad (1)$$

where Y is the vector of M ground truth target images each with N pixels, Y_{ij} is the jth pixel of the ith image, and $\hat{Y}$ is the vector of predicted images. Each image Y_i has a 3×3 block showing the true location of the transmitter, with a value of 0.01 on the outer edges and 0.92 in the center

Fig. 4. Plots of model outputs for CoM, PSE, and EMD (l-r). The CoM outputs are non-interpretable and do not provide any model confidence values.

pixel. We use the 3×3 block instead of a single pixel to prevent normalization issues and provide model confidence scores. We refer to this loss as pixel squared error (PSE). One key weakness of PSE is that since it is computed separately for each pixel, the loss does not capture the relationship between the predicted location and the actual transmitter location; in other words, if a prediction is 5 pixels from the true location, this receives the *same penalty* as a prediction 50 pixels away. The PSE objective is to approximate an image, and it does not directly minimize the localization error; this results in difficulty training models and requires a careful tuning of learning rates and model architectures to achieve low error.

Results on which formulation is more accurate have been mixed: our work in [7] found the target image setting to be more effective than regression settings, though Yapar et al. [9] found their regression formulation to be more accurate on simulated data. This indicates that metaparameters such as the model architecture, input resolution, or training data characteristics may determine which loss function is preferred.

The regression setting in [9] uses the center of mass of the output image as the predicted coordinates. However, this "center of mass" (CoM) loss function can only be used for a single prediction, and it also produces non-intuitive, uninterpretable results, as shown in Fig. 4 (left). The target image settings on the left and right produce point predictions indicating the probable transmitter location, but the CoM prediction has no such point prediction. Although interpretable results may not be a requirement in every setting, later in this work, we show how interpretable point predictions can be used to indicate model confidence and improve robustness.

In summary, there are two primary problems with existing loss functions and formulations:

1) Regression settings fail to provide interpretable results or any score indicating model uncertainty.
2) Target Image settings do not minimize localization error in the loss functions, making models challenging to train and often resulting in lower accuracy.

B. Earth Mover's Distance as a Loss Function

We propose using the Earth Mover's Distance (EMD) [12], also known as the Wasserstein-1 metric, for training a localization model. Intuitively, the EMD represents the amount of "work" that must be completed to move a pile of sand with size and location given by P to fill a hole with size and location given by Q. For probability distributions, EMD represents the "work" to change distribution P to match distribution Q.

Given two signatures composed of locations and values, $P = \{(x_1, p_1), \ldots, (x_m, p_m)\}$ and $Q = \{(y_1, q_1), \ldots, (y_n, q_n)\}$, the EMD is defined according to some optimal flow $F = (f_{ij})$ which minimizes the work

$$W(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij} \tag{2}$$

where $d_{ij} = d(x_i, y_j)$ is the Euclidean distance between x_i and y_j. Once the optimal flow problem is solved, the EMD between P and Q is defined as

$$\text{EMD}(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f^*_{ij} d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f^*_{ij}}. \tag{3}$$

where f^*_{ij} is the optimal flow.

In the localization context, our ML model learns to approximate the 2D distribution marking the true transmitter location. Using EMD to train localization models addresses the two problems with existing loss functions. Unlike the PSE setting, EMD captures the geometry of localization by placing a larger penalty on nonzero values that are farther away from the true location, which improves accuracy and ease of training. And unlike regression settings, EMD also provides an interpretable output, as shown in Fig. 4.

However, calculating the EMD in multiple dimensions is computationally expensive, with a computational complexity in $\mathcal{O}(n^3 \log n)$, where n is the number of pixels. To reduce this complexity, we use the *sliced EMD* from [20], which takes the average EMD from random 1D-projections or *slices* of the distribution and approximates the 2D EMD. With the differentiable optimal transport library, Python Optimal Transport [21], we can use the sliced EMD as a loss function to train our localization model.

C. Evaluation

To motivate the use of EMD, we compare the localization accuracy of models trained with different loss functions. We compare against the PSE loss used in [7], as well as the CoM regression formulation from [9], where the predicted coordinates are the center of mass of the CNN output. We do not evaluate other regression settings since our previous work [7] showed that other regression settings were less accurate than the PSE loss using the same dataset as in this work. We find that EMD is effective as a loss function, especially when there is a distribution shift between the training set and the test data.

For each of the three loss functions, we train an ensemble of five UNet localization models, as described in [7] (a similar model architecture was also used in [9]). We use an input resolution of 30 meters per pixel, which previous work found as ideal for this dataset. Although higher-resolution inputs can improve accuracy, we found that learning at higher resolution increases training time dramatically and also requires changes to the model architecture, since a deeper network is necessary for the impact of sensors to be captured at higher resolution. Training at higher resolution also provides diminishing returns on accuracy in terms of the computation required. We use a

TABLE II
MEDIAN ERROR FOR CENTER OF MASS (COM), PIXEL SQUARED ERROR (PSE), AND EARTH MOVER'S DISTANCE (EMD)

Loss Function	Loss Type	Random Error [m]	July→Nov Error [m]	Nov→July Error [m]	Grid2 Error [m]	Grid5 Error [m]	Training Time	Multi Tx?	Interpretable?
CoM [9]	Regression	**35.5**	149.7	265.3	334.0	154.1	2.6 s/epoch	No	No
PSE [7]	Target Image	95.4	212.6	298.0	303.2	186.2	**2.5 s/epoch**	Yes	Yes
EMD	Target Image	47.4	**116.5**	**234.2**	**277.5**	**147.6**	14.3 s/epoch	Yes	**Yes**

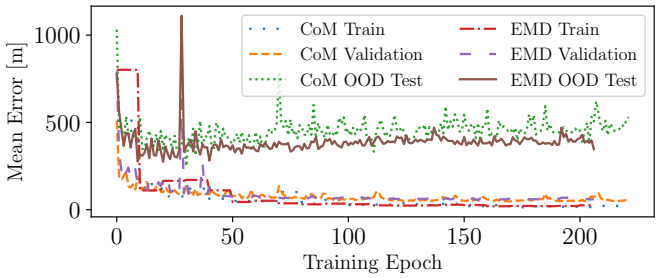

Fig. 5. Training accuracy for a single network on the Grid2 data, comparing CoM and EMD. EMD is consistently more accurate on OOD Test samples.

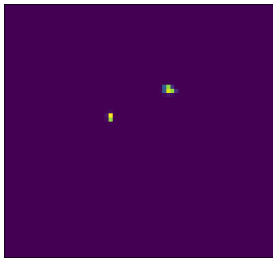

Fig. 6. An example of an ambiguous prediction from an EMD-trained model.

constant learning rate of 5×10^{-4} with the Adam optimizer [22]. We train the ensemble for a variable number of epochs; training halts after 200 epochs without decresing error on the validation set, and the best performing model on the validation set is saved. We report the median accuracy from the trained models in Table II, along with information on the loss function and training time.

To demonstrate the training stability for CoM and EMD, we compare train, validation, and OOD test accuracy during training in Fig. 5. We show the accuracy for a single network rather than an ensemble of five. We do not show PSE here as the training is unstable, often requiring restarting the training process with different learning rates. As we can see in both Table II and Fig. 5, EMD is consistently more accurate on the OOD test data.

For the best-case *Random* data, CoM is significantly more accurate. This is an expected result since the Target Image settings make predictions for a specific pixel, while the CoM setting can predict with sub-pixel precision. Without sub-pixel predictions, we expect EMD error to be 7.5 m higher on average. Accounting for this error, there is still a minor advantage (<5 m) for CoM. Again, the *Random* separation is the best-case scenario for ML models and is quite unrealistic in practice. In the other OOD settings, EMD is more accurate by 2.5-22% even without sub-pixel predictions.

As shown in Fig. 4, EMD and PSE produce interpretable outputs, with the probable transmitter location clearly marked. Conversely, CoM produces irregular shapes that do not clearly show the transmitter location. More importantly, the CoM formulation does not provide flexibility in predictions. For example, EMD and PSE can allow for some degree of ambiguity (the transmitter is either in location A or location B), which can be crucial information for enforcing spectrum usage since localization with low sensor density may be an ill-poised problem, depending on the positions of sensors and transmitters. Fig. 6 shows an example of an ambiguous prediction. Target Image models can also be extended to localize multiple simultaneous transmitters (as in [10], [18]). This is not possible with the CoM regression loss. Although our dataset consists of only single transmitter samples, we consider EMD's ability to provide multiple predictions to be a significant advantage.

Regarding training time, EMD takes significantly longer during the backward training pass, though this is of less concern since training is a one-time cost. Both CoM and EMD can take over 1000 epochs to finish the training process, while PSE typically halts learning after less than 300. This early stopping, along with the poor accuracy, demonstrates that PSE models are difficult to train and may require careful metaparameter tuning to achieve higher accuracy, while the other loss functions are more stable during training.

In summary, we find that the EMD loss function is preferred. The CoM loss function may be superior in specific circumstances due to better accuracy on in-distribution data or when training time is a concern, but the EMD formulation is more robust on OOD samples. Most importantly, EMD also naturally learns to estimate model confidence. In the following sections, we use this model confidence to identify OOD samples and improve model robustness.

IV. USING MODEL CONFIDENCE TO UNDERSTAND AND IDENTIFY FAILURE IN ML LOCALIZATION

Prediction on OOD samples remains a significant challenge for any localization model. This section explores using *model confidence* to understand cases of high error and identify OOD samples. We explore different indicators of model confidence and find the maximum value in a prediction to be the best indicator of confidence. We investigate samples with high error and low confidence to understand why localization fails, and we show that confidence can be used to identify OOD samples.

In this section, we consider two main types of distribution shift: the *Grid* setting where training data does not include transmitter locations from parts of the region of interest, and the *Seasonal* change where one data partition was collected during sunny July, and the other partition was collected during a winter storm in November. These were both described in Section II-B.

A. Indicators of Confidence

When a model encounters data that differs significantly from the training distribution, we expect uncertainty to arise. Model confidence is used in this work to identify such uncertainty. Ideally, when a model has been trained on a specific sample, it should have high confidence in the predicted location. Most loss functions, including our EMD formulation, will penalize predictions in incorrect locations, so model confidence is naturally learned during the training process, and we expect lower confidence when an input is significantly different from any that the model has been exposed to during training. We attempt to use confidence as a meta-signal, indicating either an OOD sample or a high-error, in-distribution sample.

In classification tasks, ML models predict a value for each possible class, and these values are often interpreted as model confidence for each class [23]. In our localization context, we do not have such a convenient confidence score, so we explore several options here. One option is using the maximum value in a prediction as the model confidence. However, our early experiments showed that the value of the maximum prediction is not stable between models, so we use an ensemble of 5 identical localization models, which are separately trained and then combined at inference time. The ensemble prediction is the weighted average of the maximum pixel location from each ensemble prediction.

In our localization context, we consider the following indicators of model confidence:

- *Maximum Prediction:* The maximum predicted value from any ensemble prediction.
- *Maximum Mean Prediction:* The maximum predicted value from the mean of the ensemble predictions.
- *Maximum Spread:* The maximum distance between the maximum pixel from each ensemble prediction.
- *Sum Spread:* The sum of distances between the maximum pixel from each ensemble prediction.

To determine which indicator is most effective, we use model confidence to identify OOD samples. We evaluate the model on our validation set, which consists of samples not seen during training but drawn from the training distribution. We then define a confidence threshold using the kth percentile of the indicator values from the validation set. Any samples with confidence below the threshold are labeled as "low-confidence".

In Fig. 7, we compare the effectiveness of these different indicators of model confidence at identifying OOD samples. The main finding is that the maximum mean prediction is most useful as an indicator of model confidence. This is expected since maximum mean prediction captures the information contained in the other indicators, including the confidence of individual predictions as well as the spread of the ensemble. For the rest of this work, when we refer to model confidence, we specifically are referring to *Maximum Mean Prediction*.

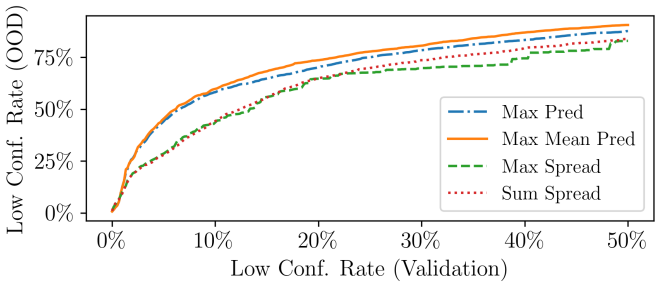

Fig. 7. ROC curve, comparing the ability of indicators of model confidence to identify OOD samples in the *Grid2* setting.

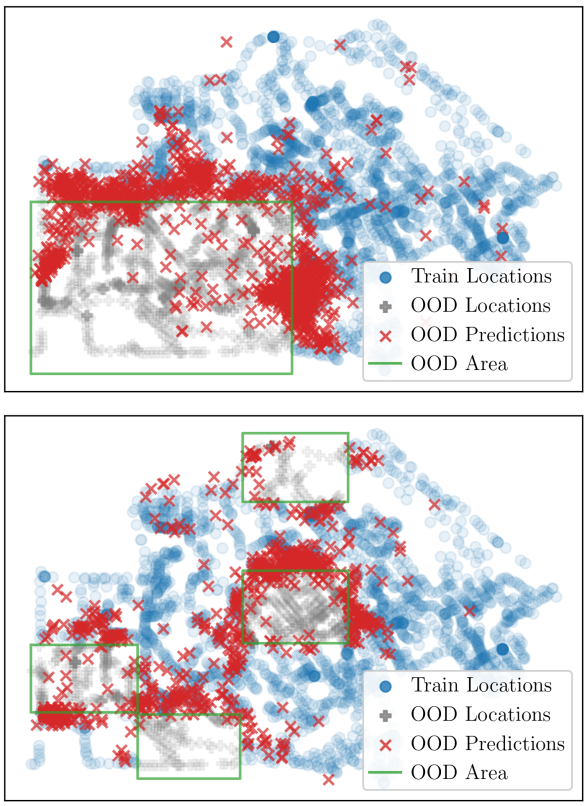

Fig. 8. Predictions for OOD targets in the *Grid2* (top) and *Grid5* (bottom) settings.

B. Understanding Localization Failure

Before considering how model confidence relates to localization accuracy, we must first understand what typical predictions from our model look like and how exactly they fail in the case of OOD predictions. We train models for localization in the *Grid2* and *Grid5* settings and plot a map of all predictions on the test sets, which is shown in Fig. 8. Each red 'X' marks a model prediction, with each OOD area

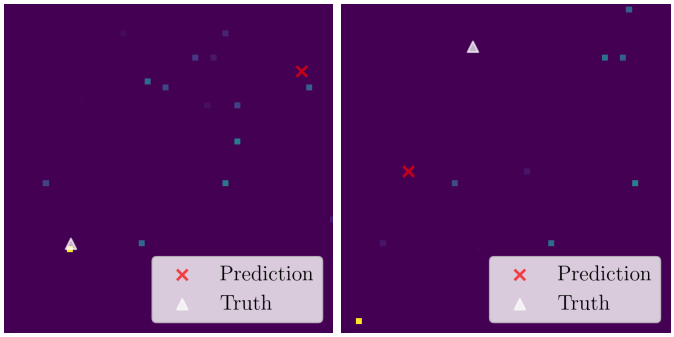

Fig. 9. Example inputs with ground truth and prediction marked on a *Grid5* OOD input (left) and validation sample showing possible interference (right).

outlined in green. In general, model predictions are made on the border of OOD regions rather than inside the area. This indicates that, for the most part, localization models predict coordinates similar to those seen during training. This is expected mainly due to the data-based nature of ML models.

The trends in model predictions highlight a significant weakness of this ML localization technique: the localization predictions are similar to those seen during training, despite highly informative sensor readings. This is shown very clearly in the example in Fig. 9 (left), which shows the input RSS values for an OOD *Grid2* sample. The true transmitter location, indicated by the white triangle, is next to a high-valued RSS sensor, shown in yellow. The model's inaccurate prediction, shown by the red X, is far from the truth even though RSS values clearly indicate the correct location. Understanding exactly what causes high error for OOD samples can allow us to design correction measures, as we will show later in Section V.

With the understanding that models are heavily biased to make predictions in previously seen locations, we can now use model confidence to study failure cases. We manually investigated the inputs for the 20 lowest-confidence predictions in the *Grid2* and *Grid5* settings and found that in both validation sets, approximately 50% of the lowest-confidence predictions had high-valued RSS sensors far away from the actual transmitter location, indicating interference from other devices or from unusually strong multipath interference. An example of this is shown in Fig. 9 (right).

The ability to identify samples with unusual RSS patterns using model confidence is one major advantage of our ML technique. In practice, a DST or other spectrum management system might raise flags asking for expert input or could repeat RSS measurements for additional information.

C. Detecting OOD Samples

As we demonstrated in [7], ML localization methods cannot accurately localize a transmitter in OOD settings. A natural question is if the model can self-identify these OOD samples that have poor accuracy. We conducted a simple experiment to determine if model confidence is an indicator of OOD inputs. For each of the data separations, we trained our 5-member CNN ensemble for localization using EMD and evaluated the model on the in-distribution validation set, as well as the corresponding OOD test set.

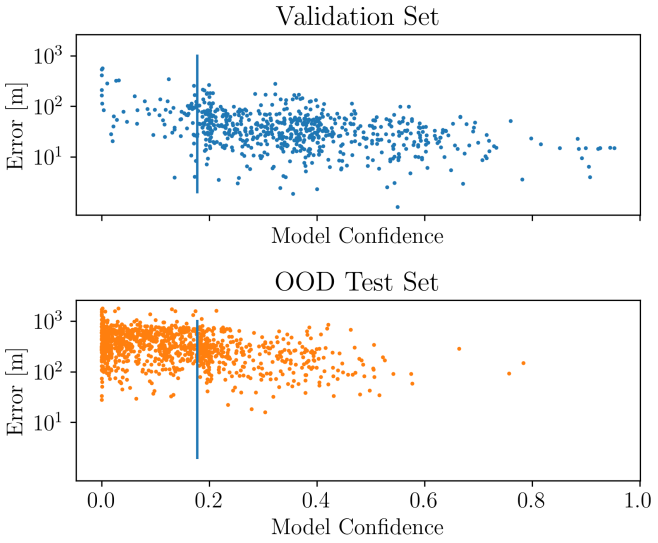

Fig. 10. Scatter plots of error vs. model confidence for the in-distribution validation set (top) and OOD test set (bottom) in the *Grid2* setting. A vertical line marks the 10th percentile of the validation set confidence.

In Fig. 10, we compare the model confidence and localization error for each prediction in the *Grid2* setting, with the validation set results on top and the OOD results on bottom. We can see that the OOD samples have much higher error and lower confidence. The relationship between confidence and error is visible but not strong, with Spearman correlation coefficients of -0.40 and -0.32 for the validation and OOD sets, respectively.

We use confidence values from the validation set to establish a "low-confidence" threshold, which allows us to classify difficult samples. The vertical lines drawn in Fig. 10 are at the 10th percentile of confidence values in the validation set. The 10th percentile is near the true positive/false positive "knee" shown in Fig. 7. A lower percentile can be used, but this fails to identify most of the OOD samples, and a higher percentile would classify more of the validation set as low-confidence.

Using the 10th percentile threshold, we identify 66.6% of the OOD samples as low-confidence. 99% of these samples have higher error than the validation set median error, and 98% are more than 2× the validation median error. Of the 10% of low-confidence predictions in the validation set, 78% of these predictions have an error higher than the median error, and 54% have more than 2× the median error. In other words, low-confidence samples typically have high error, and model confidence is a reasonable indicator of prediction accuracy for OOD and other difficult samples.

Table III shows the rate of OOD samples detected as low-confidence, along with the median error of the high-confidence OOD predictions, low-confidence OOD predictions, and in-distribution predictions. The low-confidence rate represents the

TABLE III
LOW-CONFIDENCE RATES FOR DIFFERENT OOD SETTINGS

Setting	Low-Conf Rate (OOD)	Median Error [m] High-Conf	Low-Conf	In-Distr.
Grid2	66.6%	231.3	397.5	34.6
Grid5	45.4%	134.0	216.4	32.8
July	40.8%	117.6	115.7	55.1
November	72.8%	84.8	308.6	61.2

effectiveness of model confidence at identifying OOD samples.

As expected, the low-confidence rate is significantly lower in *Grid5* than in *Grid2* (45.4% vs. 66.6%), since the *Grid5* setting has less separation between the training and OOD sets. In practice, we expect that no classifier should achieve perfect accuracy at identifying OOD samples. In this *Grid* setting, samples are arbitrarily separated from the training set, with some points very close to the training data. For points near this boundary, no classifier should be able to distinguish these from the training distribution so these samples will not be identified as low-confidence.

For the *Seasonal* separations, the *July* model was trained on data from July and evaluated on data from November, and vice versa for the *November* model. The July model has a relatively low rate of low-confidence on OOD samples compared to November (40.8% vs 72.8%). This is likely due to the July data having $10\times$ the number of transmitter locations and covering most of the campus, while the November data only covers a small portion.

Similar to the pattern observed in the *Grid* case, a higher low-confidence rate indicates a larger distribution shift, meaning the small set of November data is significantly different from the more diverse set of July data. This idea is also supported by the fact that high-confidence OOD samples in the November setting had the closest accuracy compared to the in-distribution samples. This indicates that the November model has specialized on a small set of data, and any high-confidence predictions closely resemble that data. On the other hand, the July training data is far more diverse, so the model is less effective at identifying OOD samples, resulting in both low rates of OOD detection, as well as high error for high-confidence OOD samples.

Regarding spectrum management, the low-confidence label can provide helpful context for decision-making. For example, when an RDZ management system is presented with low-confidence predictions, resources such as crowdsourced or mobile sensors can be activated within a local area for improved accuracy. Additionally, if information on the transmitter location can be obtained, low-confidence samples can be incorporated into training data to continuously maintain the localization model.

V. HYBRID LOCALIZATION USING MODEL CONFIDENCE

Although detecting OOD samples using model confidence does provide helpful context to a DST or other spectrum management system, ideally an accurate localization prediction can still be made. We consider hybrid localization models, where the DST can use an alternate technique for localization in case of low-confidence predictions.

In this section, we perform two case studies on hybrid localization models. First, we use a path loss-based localization approach as a fallback for low-confidence predictions. Path loss-based approaches generally have much higher error since they do not capture environmental information such as buildings and other obstacles, but this means they are also not biased toward specific locations like CNN-based techniques. We show that in cases of significant distribution shift, the fallback predictions can drastically improve accuracy on OOD samples, with minimal impact on in-distribution predictions.

We then explore how confidence from different ML models can be used to select the more accurate prediction. Pretrained ML models are often shared in a model zoo or commons. For localization on novel samples from unknown distributions, model confidence can be used to determine which model is most appropriate for use on a given sample.

A. RSS Ranging Localization as a Fallback

Signal strength ranging methods use a propagation loss model to estimate the distance between a sensor and a transmitter. In this work, we choose the classic log-distance path loss model [24] as a simple example to estimate the distance. Then we use a Min-Max heuristic, as outlined in [25], to estimate the transmitter coordinates. In comparison to ML techniques, this method is extremely noisy as the path loss model cannot capture the effect of the environment and multipath on the propagation loss.

We tested several of the propagation models proposed in [25], and selected the 3GPP Macro-cell log-distance path model:

$$d_i = 10^{\left(\frac{-r_i + a}{37.6}\right)}, \tag{4}$$

where r_i is the observed RSS value, d_i is the estimated distance between the transmitter and sensor i, and a is an RSS calibration factor learned from the training data. We then use a Min-Max heuristic to estimate the transmitter coordinates. Given a set of sensor locations and distances from each sensor, we find the maximum of all lower-bound estimates, the minimum of all upper-bound estimates, and take the average of the minimum and maximum as our estimated coordinate. Although least-squares approaches such as in [26] are commonly used for localization, during our experiments and in [25], this resulted in high localization error compared to the Min-Max heuristic.

It should be noted that our path loss model and ranging heuristic are not the only options for a fallback. Different propagation models can be used, and more complex localization techniques that include details about the environment can improve the hybrid model's accuracy. In this work, we intend to illustrate a generic framework for hybrid localization in the DST which could be extended through more sophisticated propagation models or heuristics.

Fig. 11 shows the median error of the CNN, ranging, and hybrid localization methods for the in-distribution and OOD

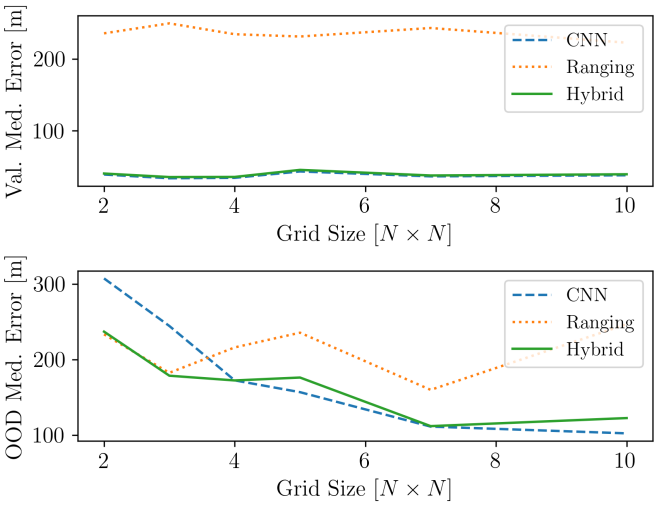

Fig. 11. Comparison of median error vs. grid size for the CNN, ranging, and hybrid models on the in-distribution (top) and OOD (bottom) data. A smaller grid size represents a larger distribution shift.

sets at different grid sizes. The in-distribution accuracy (top) remains constant for the CNN predictions as the grid size increases, while the OOD error (bottom) decreases with the grid size. This is due to the reduced separation between the training and test distributions as the grid size increases. Our hybrid model, which uses ranging localization when CNN predictions have low confidence, is nearly identical to the CNN for the in-distribution set, since few samples have low confidence. For the OOD samples, our hybrid model has median accuracy between the ranging and CNN models, and this dramatically improves the accuracy of predictions on OOD samples when the grid size is low.

However, this hybrid technique is not always preferred. Since ranging localization has a relatively high error, it only improves on the CNN baseline at low grid sizes, or in other words, when a very large distribution shift occurs. If we suppose the expected OOD error can be estimated, for example, by measuring the size of areas that are not covered by the training set, then spectrum managers can determine if a fallback technique such as this is appropriate in a region, since the median error for the fallback technique is similar for in-distribution and OOD samples.

Alternatively, the simple ranging approach we use here can be replaced by a more accurate physics-based localization model that accounts for environmental obstacles, but developing such a model is beyond the scope of this work.

B. Using Confidence from Multiple ML Models

Now, we consider the case where multiple ML models provide predictions and a model confidence score with each prediction. The DST can access multiple localization models in our setting, each with a known threshold indicating low-confidence. In deployment, these models encounter samples from an unknown distribution, so the most confident prediction will be used since it is unknown which model will be more accurate. Since confidence values are only relative to one specific model, we divide all values by the low-confidence threshold to provide a normalized model confidence. After normalization, low-confidence values are less than 1 and high-confidence values are more than 1, and we can compare confidence between the two models to select the better prediction.

TABLE IV
RESULTS FROM 2-MODEL HYBRID LOCALIZATION ON SEASONAL DATA

Data	July Model Median [m]	Nov Model Median [m]	Hybrid Model Median [m]	Hybrid Model Best Choice Rate
July	55.1	234.2	55.8	87.4%
Nov	116.5	61.2	94.0	57.7%
April	400.6	390.0	391.7	47.2%

In our experiment, two models are trained on the data from *July* and *November*, and our test set is made up of random samples taken in July and November, with a small set taken in April, which has significantly fewer RSS sensors. Table IV shows the results from this experiment. We evaluate the two models on each test set, and then consider our "hybrid model", where the prediction with the higher normalized confidence is used. We show the median error for each model, as well as the "Best Choice Rate", or the rate at which the hybrid model chooses the best choice from the two predictions.

As expected, the hybrid model is less accurate than each model on its in-distribution data, but it provides a reasonable balance of accuracy between the two single models. The hybrid model makes the best choice over 87% of the time on July data but on the November data, there is a much lower success rate of 57.7%. We expect that this lower rate is due to the more limited set of training data used by the November model. The July training data is much more diverse in terms of locations covered, so it is often more confident on the test samples from November.

Note that for the July and November data, the best choice rate of the hybrid model is significantly higher than the OOD detection rates in the previous section. Since our hybrid model uses two normalized confidence scores to make a decision, it is ultimately more sensitive than OOD detection which only uses one confidence score.

The April data is OOD for both models, so we see that the rate of choosing the best choice is close to random, and both trained models have similarly poor performance. This highlights that normalizing our confidence scores is having the intended effect. For the OOD April inputs, confidence is equally low for both models, so one model is not preferred more than the other.

In summary, we can effectively combine predictions from multiple models to achieve lower error by utilizing normalized confidence. These results validate the general framework of model confidence we have established by showing that confidence is a reasonable indicator of which training distribution a sample resembles and can be used to improve overall accuracy.

VI. Conclusion

In this work, we considered how interpretable predictions with accompanying confidence scores can be used for more effective ML localization systems. We proposed using Earth Mover's Distance as a loss function for training localization models. Training models with EMD addressed the primary challenges of existing loss functions by providing greater accuracy, interpretable results, and a more stable and robust training process. EMD improves accuracy on OOD samples by up to 22% compared to other techniques.

Using EMD also allowed us to explore model confidence, which can be used to identify high-error and OOD samples, providing valuable context for decision-making in spectrum management. We found that model confidence can help find pathological samples with unexpected behavior such as third-party interference.

We explored how the context provided by model confidence can be used for hybrid localization. We gave two examples of how confidence can be used in utilizing predictions from different models to provide robustness against OOD samples. We considered a physics-based fallback approach that improves accuracy on OOD samples in case of large distribution shift, and we showed how predictions from multiple ML models can be compared in terms of normalized confidence, allowing the DST to choose the more accurate prediction.

Our goal is to integrate model confidence as a fundamental aspect of localization systems. Future research will focus on a deeper exploration of hybrid models and fallback techniques, as well as deploying our methods to inform decision-making in a zone management system.

References

[1] T. Kidd, "National radio quiet and dynamic zones," in *Information Technology Magazine*, The Department of the Navy, 2018.

[2] S. J. Maeng, I. Güvenç, M. Sichitiu, B. Floyd, R. Dutta, T. Zajkowski, O. Ozdemir, and M. Mushi, "National radio dynamic zone concept with autonomous aerial and ground spectrum sensors," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 687–692, IEEE, 2022.

[3] M. Zheleva, C. R. Anderson, M. Aksoy, J. T. Johnson, H. Affinnih, and C. G. DePree, "Radio dynamic zones: Motivations, challenges, and opportunities to catalyze spectrum coexistence," *IEEE Communications Magazine*, pp. 1–7, 2023.

[4] G. D. Durgin, M. A. Varner, N. Patwari, S. K. Kasera, and J. Van der Merwe, "Digital spectrum twinning for next-generation spectrum management and metering," in *2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI)*, pp. 1–6, IEEE, 2022.

[5] S. Tadik, K. M. Graves, M. A. Varner, C. R. Anderson, D. Johnson, S. K. Kasera, N. Patwari, J. Van der Merwe, and G. D. Durgin, "Digital spectrum twins for enhanced spectrum sharing and other radio applications," *IEEE Journal of Radio Frequency Identification*, 2023.

[6] M. Arnold and M. Alloulah, "Benchmarking learnt radio localisation under distribution shift," *arXiv preprint arXiv:2210.01930*, 2022.

[7] F. Mitchell, N. Patwari, S. K. Kasera, and A. Bhaskara, "Learning-based techniques for transmitter localization: A case study on model robustness," in *20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2023.

[8] F. Mitchell, P. Smith, A. Bhaskara, and S. K. Kasera, "Exploring adversarial attacks on learning-based localization," in *Proceedings of the 2023 ACM Workshop on Wireless Security and Machine Learning*, pp. 15–20, 2023.

[9] Ç. Yapar, R. Levie, G. Kutyniok, and G. Caire, "Real-time outdoor localization using radio maps: A deep learning approach," *IEEE Transactions on Wireless Communications*, 2023.

[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*, vol. 82, p. 101582, 2022.

[11] 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.

[12] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, pp. 99–121, 2000.

[13] S. Tadik, M. A. Varner, F. Mitchell, and G. D. Durgin, "Augmented rf propagation modeling," *IEEE Journal of Radio Frequency Identification*, vol. 7, pp. 211–221, 2023.

[14] P. Smith, A. Luong, S. Sarkar, H. Singh, N. Patwari, S. Kasera, K. Derr, and S. Ramirez, "Sitara: Spectrum measurement goes mobile through crowd-sourcing," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 46–54, IEEE, 2019.

[15] J. Talvitie, E. S. Lohan, and M. Renfors, "The effect of coverage gaps and measurement inaccuracies in fingerprinting based indoor localization," in *International Conference on Localization and GNSS 2014 (ICL-GNSS 2014)*, pp. 1–6, IEEE, 2014.

[16] T. Janssen, R. Berkvens, and M. Weyn, "Benchmarking rss-based localization algorithms with lorawan," *Internet of Things*, vol. 11, p. 100235, 2020.

[17] 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 21st International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pp. 201–210, 2020.

[18] 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.

[19] F. Mitchell, A. Baset, S. K. Kasera, and A. Bhaskara, "A Dataset of Outdoor RSS Measurements for Localization," *Zenodo https://doi.org/10.5281/zenodo.7259895*, Oct. 2022.

[20] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, "Sliced and radon wasserstein barycenters of measures," *Journal of Mathematical Imaging and Vision*, vol. 51, pp. 22–45, 2015.

[21] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer, "Pot: Python optimal transport," *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*, pp. 1321–1330, PMLR, 2017.

[24] T. Rappaport, *Wireless Communications: Principles and Practice*. USA: Prentice Hall PTR, 2nd ed., 2001.

[25] M. Aernouts, B. Bellekens, R. Berkvens, and M. Weyn, "A comparison of signal strength localization methods with sigfox," in *2018 15th Workshop on Positioning, Navigation and Communications (WPNC)*, pp. 1–6, IEEE, 2018.

[26] 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.