GNN-SML: Graph Neural Network-Based Spectrum Misuser Localization

Yan Zhang*, Tao Li†, Yanchao Zhang‡
* University of Akron, † Purdue University, ‡ Arizona State University
yanzhangyz@uakron.edu, litao@purdue.edu, yczhang@asu.edu

***Abstract*—Spectrum misuser localization (SML) is essential for dynamic spectrum sharing (DSS) to ensure that only authorized users access and utilize shared spectrum. In this paper, we introduce GNN-SML, an innovative framework for crowdsourcing-based DSS systems that accurately and simultaneously localizes multiple spectrum misusers with unknown locations and transmission powers, even when they are in close proximity. GNN-SML employs location-centric, sensor-agnostic Graph Neural Networks (GNNs) with inductive power. In each online SML instance, these GNNs predict the Received Signal Strength (RSS) residue between each potential transmitter and available spectrum sensors, which may be located arbitrarily and not involved in the training phase. These predicted RSS residues, combined with real-time RSS measurements from spectrum sensors, are used with the zero-forcing technique to estimate the locations and transmission powers of all potential transmitters. This enables the identification of spectrum misusers as localized transmitters lacking proper spectrum authorizations. We validate the effectiveness and efficiency of GNN-SML with real indoor and outdoor datasets. Compared to leading methods, GNN-SML reduces localization error by up to 58% and transmitter power estimation error by up to 24%, while maintaining a comparable localization time of 1.2 s.**

***Index Terms*—Spectrum misuser localization, GNN, dynamic spectrum sharing**

I. Introduction

Dynamic Spectrum Sharing (DSS) is a crucial solution to the global shortage of wireless spectrum. The standard DSS approach utilizes a Spectrum Access System (SAS) to issue spatiotemporal spectrum authorizations while safeguarding high-priority users and existing DSS users from interference. However, DSS systems face challenges from spectrum misusers who attempt to access the spectrum without a proper SAS authorization. The increasing availability of affordable software-defined radio (SDR) devices has intensified this threat. Consequently, spectrum misuser localization (SML) has become essential for maintaining effective DSS operations [1], [2], [3], [4]. Currently, the U.S. FCC and similar agencies in other countries rely on complaints and manual investigations to detect spectrum violations. These methods are time-consuming, labor-intensive, and costly, particularly given the mobility and adaptability of smart spectrum misusers.

SML can be seen as a specific case of wireless transmitter localization, a well-established field. If a localized transmitter lacks an active spectrum authorization, the SAS operator can classify it as a spectrum misuser. Traditional transmitter localization techniques often employ a network of sensors for spectrum monitoring [5]. These sensors may include dedicated devices deployed by the SAS operator and crowdsourced devices aimed at enhancing spectrum sensing coverage and accuracy. Research such as [6] has demonstrated the technical feasibility of crowdsourced spectrum sensing, a concept widely supported in the literature [7], [8], [3], [1], [2], [4]. Spectrum sensors can measure various parameters, including received signal strength (RSS), angle-of-arrival, and time-of-arrival. RSS-based localization is particularly prevalent due to its lower sensor requirements [3], [1], [2], [4]. However, traditional RSS-based methods typically focus on locating a single transmitter and assume that each RSS measurement corresponds to only one transmitter. In practice, when multiple unauthorized and/or authorized DSS users share the same frequency band, their overlapping transmissions result in aggregated RSS measurements that are challenging to disentangle. This situation underscores the need for advanced RSS-based techniques capable of *localizing multiple simultaneous transmitters with unknown locations and powers within the same frequency band*, a problem known as Multiple Transmitter Localization (MTL) [3], [4], [1], [2], [9], [10].

State of the Art. Recent years have seen notable efforts in MTL. SPLOT [3] reformulates the MTL problem into multiple single-transmitter localization tasks, identifying approximate regions for each transmitter based on local maxima in spatially distributed RSS measurements. As analyzed in [9], SPLOT can produce high false alarm rates under complex radio propagation conditions that generate many local maxima. LLOCUS [4] improves upon SPLOT by adaptively selecting these approximate regions. As noted in [3], [4], both SPLOT and LLOCUS are less effective when transmitters are very close to each other. MAP* [1], [2] employs a hypothesis-driven Bayesian approach, where each hypothesis represents a configuration (location and power) of potential transmitters, and localization involves identifying the most likely hypothesis. MAP* suffers from high computational costs due to too many possible configurations (or hypotheses) and often results in high false positives and negatives [10].

There are also deep learning-based MTL methods. DeepTxFinder [9] uses separate convolutional neural network (CNN) models for each possible number of transmitters, resulting in high model complexity and computational costs. DeepMTL in [10] and [11] use CNN models to map RSS-constructed images to transmitter distribution images and then uses this mapping to infer transmitter locations. Like MAP*, these models relies on exhaustive search for both transmitter

locations and power, leading to significant computational costs during training and inference. Moreover, converting sensor coordinates to an image has challenges, such as the impact of pixel scale on localization accuracy, where poor scaling can lead to significant errors. [12] proposes using quantum sensors to encode signal information into quantum states and localize transmitters through state transformations, similar to SPLOT's subproblem division but with scalability limited by sensor availability. [13] models localization as a supervised learning task using RSS measurements and ground-truth transmitters, constrained to specific transmitter locations and sensor setups.

Our Contributions. In this paper, we present GNN-SML, a novel framework that uniquely integrates Graph Neural Networks (GNNs) [14] with the zero-forcing technique [15] to achieve *simultaneous*, *accurate*, and *fast* localization of multiple concurrent spectrum misusers, regardless of their physical proximity. Like previous approaches [3], [4], [1], [2], [9], [10], GNN-SML essentially tackles the MTL problem and classifies any localized transmitter lacking a proper SAS authorization as a spectrum misuser.

We first discuss the rationale for applying zero-forcing [15] to MTL. Zero-forcing is an effective technique for mitigating interference in MIMO systems. In a MIMO system with M transmit antennas and N receive antennas, each receiver captures the combined power from M simultaneous transmissions. Given an $N \times M$ channel matrix $\mathbf{H}$ that characterizes the MIMO channel, zero-forcing computes a matrix $\mathbf{W}$ such that $\mathbf{WH} \approx \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. By applying $\mathbf{W}$ to the N-dimensional vector of received signals, we can effectively estimate the M transmitted signals [15]. Suppose the SAS employs N sensors (i.e., receivers) to localize transmitters over a large area divided into M equally-sized cells. Assume that a potential transmitter could be located at the center of each cell, with its transmission power unknown. If the channel matrix $\mathbf{H}$ is known in this context, zero-forcing can be applied to simultaneously estimate the transmission power of each potential transmitter. If the estimated power of a potential transmitter exceeds a predetermined system threshold (e.g., the FCC power limit), it can be identified as an actual transmitter. The accuracy of transmitter power and location estimations can be refined by adjusting the cell size, which is equivalent to changing M.

The main challenge in adopting the zero-forcing method is obtaining the necessary channel matrix $\mathbf{H}$. Radio propagation models commonly used to estimate $\mathbf{H}$ are not known a priori and must be learned from real RF data. Additionally, as we assume crowdsourcing-based SAS operators, as in previous work [3], [4], [1], [2], [9], [10], the spectrum sensors involved in different MTL instances can vary dynamically in both number and location. This variability prevents the pre-determination of $\mathbf{H}$ for each cell center and fixed sensor location, necessitating the rapid derivation of $\mathbf{H}$ to enable fast MTL in each instance.

GNN-SML addresses these challenges in three steps. First, it uses an initial set of sensors to learn a fine-grained radio propagation model specific to the sensing area. Next, the same sensors are employed to train a *location-centric* and *sensor-agnostic* GNN model with *inductive power*. This model captures the spatiotemporal data dependencies among arbitrary sensing locations, regardless of whether a real sensor is present. The GNN is designed to predict the *RSS residue* between any potential transmitter and sensor locations, which represents the difference between the actual RSS measured at a sensor and the expected RSS based on the propagation model. Finally, in each online SML instance, the SAS operator uses the trained GNN model to predict the RSS residue between each cell center and available sensors, which may be at arbitrary locations and not involved in the training phase. These predicted RSS residues are then used to estimate the channel matrix $\mathbf{H}$. The zero-forcing method is subsequently applied to estimate the transmission power at each cell center, enabling the localization of potential transmitters (spectrum misusers).

We evaluate the accuracy and efficiency of GNN-SML using real datasets from both indoor and outdoor environments [2]. Our findings demonstrate that GNN-SML reduces localization error for transmitters (including potential spectrum misusers) by up to 37% indoors and 58% outdoors compared to state-of-the-art methods [3], [4], [1], [2], [9], [10]. Additionally, GNN-SML lowers transmitter power estimation error by up to 20% indoors and 24% outdoors. Furthermore, GNN-SML achieves a localization time of 1.2s on a standard Dell desktop, which is comparable to the leading method but with significantly reduced localization and power estimation errors. These results affirm GNN-SML's effectiveness in accurately localizing simultaneous spectrum misusers in nearly real-time, regardless of their physical proximity.

The rest of this paper is organized as follows. §II presents the problem formulation. §III illustrates the design of GNN-SML. §IV evaluates the performance of GNN-SML. §V concludes this paper.

II. Problem Statement

A. Threat Model

This paper addresses the detection of simultaneous spectrum users, categorized into three types: unauthorized DSS users, intentional jammers, and unintentional jammers resulting from compromised wireless devices. The exact number of misusers is unknown, and they may operate with arbitrary transmission power anywhere within the SAS's sensing area. Moreover, they may be located in close proximity to one another, which complicates their localization and separation with current techniques [3], [4]. Although spectrum users can move and change their online/offline status unpredictably, their transmissions persist at a given location long enough to make localization efforts feasible and meaningful. We also assume a white-box threat model in which spectrum misusers are fully aware of SAS system operations, including our localization mechanism. Additionally, we employ existing defenses [7], [16], [17] to address the issue of compromised spectrum sensors that may submit fake sensing data.

B. System Model

Consistent with previous studies [3], [4], [1], [2], [10], [7], [18], [8], [16], we consider a crowdsourcing-aided SAS that leverages both dedicated and crowdsourced (spectrum) sensors for spectrum access control. The SAS operator conducts one round of SML either periodically or as needed. Let V denote the set of $N_v = |V|$ responsive sensors in a given SML round, with the round index omitted for brevity. Crowdsourced sensors can be recruited for specific sensing tasks and removed upon task completion. The SAS operator has the flexibility to add or remove both dedicated and crowdsourced sensors as needed. Additionally, some sensors might experience temporary unresponsiveness. Consequently, V can vary significantly across different SML rounds in the same area, which poses one of the biggest design challenges. Furthermore, since sensors primarily function as passive receivers, we use "sensors" and "receivers" interchangeably when no confusion arises.

C. Design Goals

Consider an arbitrary SML round in a given physical area, where there can be $N_u = |U|$ simultaneous transmitters on the same frequency band, denoted by U. The number N_u, as well as the location l_u and transmission power P_u of each transmitter $u \in U$, are all unknown to the SAS operator. Each sensor $v \in V$ provides a spectrum-sensing report that includes its location l_v and a sequence $\mathbf{x}_v$ of RSS measurements collected during this round.

Given the RSS measurements $\mathbf{X} = \{\mathbf{x}_v\}_{v \in V}$, the goal of GNN-SML is to *simultaneously*, *accurately*, and *quickly* estimate the transmission power P_u and location l_u of each transmitter $u \in U$. Any localized transmitter $u \in U$ not associated with an active spectrum authorization is considered a spectrum misuser by the SAS operator.

III. GNN-SML Design

In this section, we outline the GNN-SML design. As shown in Fig. 1, GNN-SML consists of an initial training phase and an online SML phase. In the training phase, the SSP uses the available sensors to learn a precise radio propagation model specific to the sensing area. It then trains a location-centric, sensor-agnostic GNN with inductive capabilities. This GNN is designed to accurately estimate the RSS residue between potential transmitter and sensor locations. In each SML round, the SAS operator employs the trained GNN to predict RSS residues between each potential transmitter location and available sensors. Zero-forcing [15] is then used to localize all transmitters based on the predicted RSS residues, identifying those without proper spectrum authorizations as spectrum misusers. We provide a detailed explanation below.

A. Learning Radio Propagation Models

We assume a short, trustworthy system initialization phase in which no spectrum misusers are present. During this phase, the SAS operator aims to develop fine-grained radio propagation models that account for path loss and large-scale shadow fading due to obstructions such as hills or high-rise buildings.

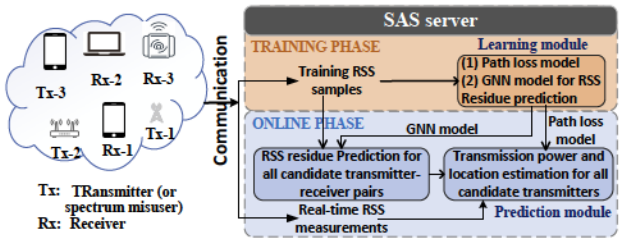


Fig. 1: GNN-SML system overview.

In this paper, we use the classical log-distance path loss model with shadow fading as a representative example. However, our approach can be easily adapted to other models. Additionally, the radio propagation model can be periodically relearned or updated as necessary to adapt to dynamic radio environments.

We use the notation V to represent the set of sensors available to the SSP during system initialization. For each sensor $v \in V$, both its location l_v and power P_v are known. The SSP divides V into two subsets: V_t and V_r. Sensors in V_t take turns acting as transmitters, while sensors in V_r continuously serve as receivers. Since only one transmitter is active at any time during the controlled training phase, the RSS measurement at each receiver is attributable to a single transmitter. Let $\mathbf{x}_{v \leftarrow v'}$ denote the RSS measurement at each receiver $v \in V_r$ induced by any transmitter $v' \in V_t$.

According to the log-distance path loss model [19], we have:

$$\begin{aligned} \mathbf{x}_{v \leftarrow v'} &= f(l_v, l_{v'}, P_{v'}) + \mathbf{s}_{v \leftarrow v'} \\ &= P_{v'} - P_0 - 10\alpha \log_{10}\left(\frac{d_{v \leftarrow v'}}{d_0}\right) + \mathbf{s}_{v \leftarrow v'}, \end{aligned} \tag{1}$$

where $f(\cdot)$ denotes the path loss model function, $d_{v \leftarrow v'} = \|l_v - l_{v'}\|_2$ denotes the Euclidean distance, α is the path loss exponent, P_0 is the path loss at a reference distance d_0, and $\mathbf{s}_{v \leftarrow v'}$ is the shadow-fading term (also known as the RSS residue). We follow the conventional assumption that $\mathbf{s}_{v \leftarrow v'}$ follows a Gaussian distribution with zero mean. To learn the model parameters $\langle P_0, \alpha \rangle$ of $f(\cdot)$, our optimization objective is to minimize the mean squared error between measured RSS values and those predicted by $f(\cdot)$. Thus, given all the collected RSS values, we formulate the optimization problem as follows:

$$\arg\min_{P_0, \alpha} \sum_{v \in V_r} \sum_{v' \in V_t} \|\mathbf{x}_{v \leftarrow v'} - f(l_v, l_{v'}, P_{v'})\|_2^2 \ . \tag{2}$$

We confirm that the log-distance path model accurately fits both indoor and outdoor datasets [2]. For brevity, detailed fitting results are omitted.

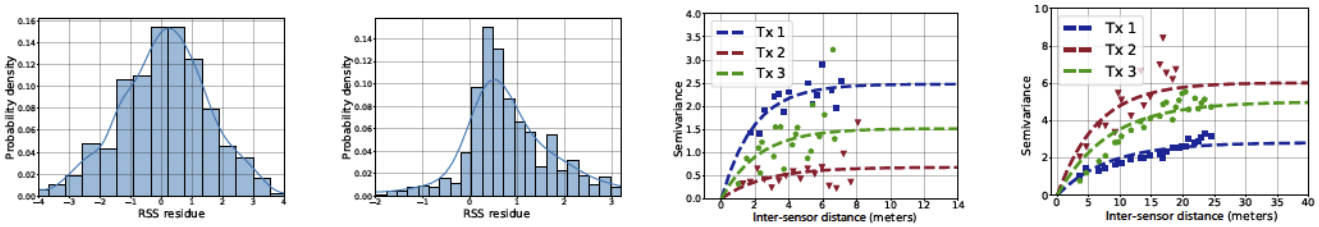
B. Learn a GNN for Predicting RSS Residues

To justify the need for learning an inductive GNN for predicting RSS residues, we first examine the role of the RSS residue defined in Eq. (1) as

$$\mathbf{s}_{v \leftarrow v'} = \mathbf{x}_{v \leftarrow v'} - f(l_v, l_{v'}, P_{v'}), \forall v \in V_r, v' \in V_t. \tag{3}$$

When Eq. (1) is expressed in Watts, it becomes

$$\mathbf{x}_{v \leftarrow v'} = P_{v'} \cdot \tilde{P}_0 \cdot \left(\frac{d_{v \leftarrow v'}}{d_0}\right)^{-\alpha} \cdot e^{\tilde{\mathbf{s}}_{v \leftarrow v'}} = P_{v'} \cdot h_{v \leftarrow v'}, \tag{4}$$



(a) RSS residues for Tx 1 in Fig. 2c (indoor) (b) RSS residues for Tx 1 in Fig. 2d (outdoor) (c) Semivariograms of RSS residues (indoor) (d) Semivariograms of RSS residues (outdoor).

Fig. 2: Spatial correlation of RSS residues at different sensors (or receivers).

where $\tilde{P}_0 = 10^{-P_0/10}$, $\tilde{\mathbf{s}}_{v\leftarrow v'} = \frac{\ln 10}{10}\mathbf{s}_{v\leftarrow v'}$, and $h_{v\leftarrow v'} = \tilde{P}_0 \left(\frac{d_{v\leftarrow v'}}{d_0}\right)^{-\alpha} e^{\tilde{\mathbf{s}}_{v\leftarrow v'}}$ represents the channel gain from transmitter v' to receiver v. In other words, given the RSS $\mathbf{x}_{v\leftarrow v'}$ and the learned model parameters (P_0, α) of $f(\cdot)$, we can directly compute $\mathbf{s}_{v\leftarrow v'}$ and subsequently $h_{v\leftarrow v'}$.

In the subsequent SML phase, one might consider determining the channel gain $h_{v\leftarrow u}$ for each potential transmitter u and sensor v using a similar approach. However, during training, each RSS measurement is attributable to only one transmitter, whereas in the SML phase, multiple transmitters may be active simultaneously. This results in an aggregated RSS measurement at sensor v that is not directly separable. Consequently, since $\mathbf{x}_{v\leftarrow u}$ is unknown, it is infeasible to derive $\mathbf{s}_{v\leftarrow u}$ for computing $h_{v\leftarrow u}$. Therefore, we need an effective method to estimate the RSS residue between each potential transmitter and the available sensor in the SML phase to apply zero-forcing.

The design of our GNN model is driven by two key observations. First, there are significant spatial correlations among the RSS residues across different pairs of transmitter and sensor locations. Second, RSS residues are primarily dependent on the locations of transmitters and sensors rather than the specific RF devices, making them location-centric and sensor-agnostic. In what follows, we first elaborate on these observations and then discuss how they inform the design of our GNN model.

1) Properties of RSS residues: Studies [20], [21], [22], [23], [24], [25] have shown that RSS measurements exhibit intrinsic stationarity. This means that the statistical properties—such as mean, variance, and covariance—of the spatial process remain consistent across different locations. This intrinsic stationarity is widely accepted in the DSS literature [26], [27], [28]. Further research [29], [30], [22] indicates that RSS residues also demonstrate intrinsic stationarity.

In particular, we model the RSS residue $\mathbf{s}_{v_i\leftarrow v'}$ associated with the sensor-transmitter location pair $(l_{v_i}, l_{v'})$ as a Gaussian random field: $\mathbf{s}_{v_i\leftarrow v'} = \mu + \delta(l_{v_i})$, where μ is the mean of the RSS residues, assumed to be zero under a Gaussian distribution, and $\delta(l_{v_i})$ represents potential sampling errors at location l_{v_i}. Consequently, We have

$$\begin{aligned} \mathbb{E}[\mathbf{s}_{v_i\leftarrow v'}] &= \mu = 0, \\ \mathbb{E}[(\mathbf{s}_{v_i\leftarrow v'} - \mathbf{s}_{v_j\leftarrow v'})^2] &= 2\gamma\left(\|l_{v_i} - l_{v_j}\|_2\right), \end{aligned} \tag{5}$$

where $\gamma(\cdot)$ represents the semivariogram function, a fundamental geostatistical tool that quantifies how the variance of differences between values of a spatial process changes with distance. This is referred to as the *receiver-location-dependent* property of the RSS residue.

Similarly, for a single sensor v at location l_v receiving RSS measurements from two different transmitters v' and v'' located at $l_{v'}$ and $l_{v''}$, respectively, we have

$$\mathbb{E}[(\mathbf{s}_{v\leftarrow v'} - \mathbf{s}_{v\leftarrow v''})^2] = 2\gamma\left(\|l_{v'} - l_{v''}\|_2\right). \tag{6}$$

We refer to this as the *transmitter-location-dependent* property of the RSS residue.

We conduct experiments using indoor and outdoor datasets [10] to validate that RSS residues exhibit strong spatial correlations dependent on both sensor and transmitter locations. Figs. 2a and 2b show that RSS residues measured at different sensor locations approximate a Gaussian distribution in both indoor and outdoor environments. Additionally, Figs. 2c and 2d present the empirical semivariograms of RSS residues in indoor and outdoor settings, involving three transmitters and various sensors. The solid lines in these plots represent fits using an exponential semivariogram model. These plots indicate that sensors in close proximity exhibit similar measurements with minimal fluctuations (i.e., smaller semivariogram values), while sensors located further apart display more pronounced variations (i.e., larger semivariogram values). Similar results demonstrating the spatial dependency of RSS residues on transmitter locations are obtained but omitted here due to space constraints.

2) GNN design: GNNs [14] are advanced deep-learning methods well-suited for analyzing spatial data. Compared to traditional geospatial interpolation, GNN-based approaches excel in modeling complex relationships, managing irregular data, and incorporating additional features while offering superior adaptability, predictive performance, scalability, and flexibility [17], [31]. Therefore, we aim to design and train a GNN to predict RSS residue between potential transmitter and sensor locations during the SML phase. A crucial feature we seek in the GNN is its *inductive* ability, which allows it to handle arbitrary transmitter and sensor locations that are unseen during training. This feature is essential because transmitters can be located anywhere within the sensing area, and spectrum sensors may vary considerably in each SML round within the crowdsourcing-aided SAS (see §II-B).

GNN Training for Unseen Sensors. We first train an inductive sensor-centric GNN to predict RSS residues for sensors not seen during training. The GNN explores latent spatial

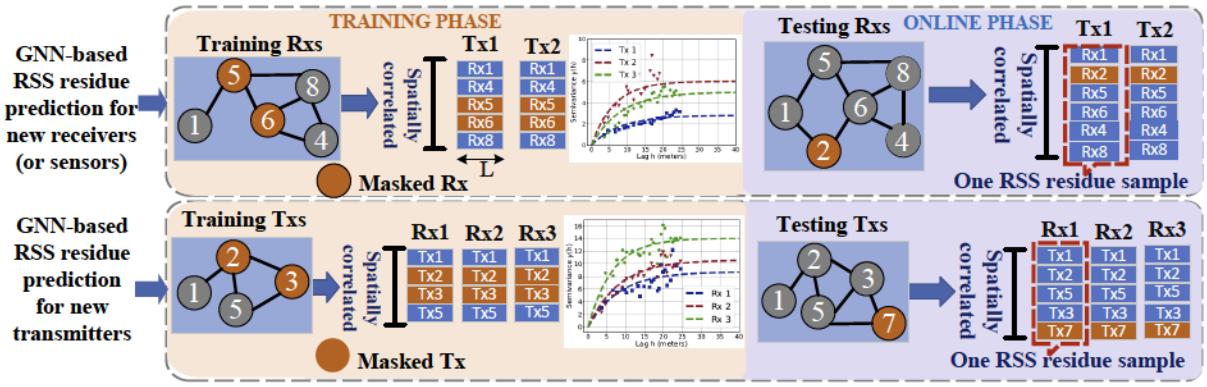


Fig. 3: Illustration of GNN training and inference.

correlations among nodes to perform interpolation and prediction of RSS residues. In this study, we adapt the Inductive Graph Neural Network Kriging (IGGNK) model proposed in [31] to our context, with the extension of our GNN-SML framework to other GNN models left as future work. Recall that the SAS operator divides the available sensors V into two subsets during training: V_t for transmitters and V_r for receivers (see §III-A). We represent V_r as an undirected graph $\mathcal{G} = (V_r, \mathcal{E})$, where each node corresponds to a sensor in V_r, and $\mathcal{E}$ denotes the set of edges connecting any two sensors within V_r. The edge weights are defined by the Euclidean distance-based adjacency matrix $\mathbf{W}$, where the element at coordinate (i, j) corresponds to sensors v_i and v_j in V_r and is given by $W_{i,j} = \exp\left(-\left(\|l_{v_i} - l_{v_j}\|_2/2\right)^2\right)$. This model characterizes the spatiotemporal dependencies among the RSS residues for each sensor in V_r, as described in Eq. (3).

We first describe how training samples are generated. During the training phase, the SAS operator instructs the transmitters in V_t to transmit sequentially, with only one active transmitter at any time. Each transmitter in V_t generates L RSS measurements at each sensor in V_r, attributable solely to that transmitter. Using these RSS measurements, the SAS operator calculates L RSS residues for each sensor in V_r as described in Eq. (3). Each sample is associated with a single transmitter and is represented as a $|V_r| \times L$ matrix, where each row corresponds to the L RSS residues for one sensor in V_r. The SAS operator can produce as many samples as needed for each transmitter. The sample generation process can be integrated with the learning of the radio propagation model.

To ensure the inductive power of the GNN, we apply a *random-masking* strategy. Specifically, we generate a new set of samples from each original sample by setting random rows in the data matrix to zero. Sensors with zeroed-out RSS residues are referred to as masked sensors, while those with nonzero residues are called unmasked sensors. Let Ω denote this new set of samples. The GNN predicts RSS residues for both masked and unmasked sensors, with masking ensuring generalization to unseen sensors during training.

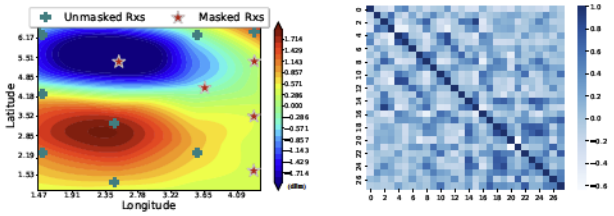
The central idea of a GNN is its use of neural message passing, where vector messages are exchanged between nodes and updated through neural networks. In our approach, message-passing operations are implemented using a three-layer GNN model, with the latent representations at each layer defined as:

$$
\begin{aligned}
H_1 &= \overline{\mathbf{W}} H_0 \theta_0^1 + (2\overline{\mathbf{W}}\overline{\mathbf{W}} - I) H_0 \theta_0^2 \,, \\
H_2 &= \sigma(\overline{\mathbf{W}} H_1 \theta_1^1 + (2\overline{\mathbf{W}}\overline{\mathbf{W}} - I) H_1 \theta_1^2) + H_1 \,, \\
\hat{S}_\omega &= \overline{\mathbf{W}} H_2 \theta_2^1 + (2\overline{\mathbf{W}}\overline{\mathbf{W}} - I) H_2 \theta_2^2 \,.
\end{aligned} \tag{7}
$$

Here, H_0 represents the input to the first layer, which is any sample $S_\omega \in \Omega$. The matrix $\overline{\mathbf{W}} = \mathbf{W}/\mathsf{rowsum}(\mathbf{W})$ denotes the normalized transition matrix derived from the adjacency matrix $\mathbf{W}$. The parameters $\{\theta_l^k \mid l = 0, 1, 2;\ k = 1, 2\}$ are learning parameters that control how each node processes the incoming data. Since the RSS residues of masked sensors are zeroed out in each sample, these sensors initially pass only zeros to their neighbors in the first layer. The second layer enables each masked sensor to transmit its layer-1 embedding to its neighbors. Given that $\mathbf{W}$ is fully connected, these two layers are sufficient to capture the spatial correlations of RSS residues among all sensors. The output of the final layer, denoted $\hat{S}_\omega$, represents the reconstructed RSS residues for both masked and unmasked sensors in the input sample. Let S_ω denote the original RSS residues for sample ω without any masking. The loss function is defined as the total reconstruction error for both masked and unmasked sensors: $\mathsf{Loss} = \sum_{\omega \in \Omega} \|\hat{S}_\omega - S_\omega\|^2$.

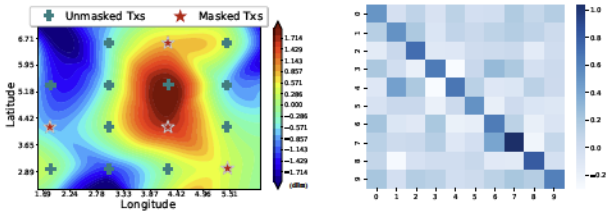
During system runtime, the trained model can be used to predict the RSS residues for sensors unseen during training, such as newly added crowdsensing devices. Specifically, we extend the graph $\mathcal{G} = (V, \mathcal{E})$ to include these new sensors, update the adjacency matrix $\mathbf{W}$, and integrate these new sensors as masked sensors into the original training samples that have not yet been subjected to random masking. We then input the modified samples and the updated $\mathbf{W}$ into the GNN, which can predict the RSS residues between each transmitter location used during training and each new sensor location. Our assumption here is that the RSS residues obtained during training remain relatively stable during the online SML phase, as they are linked to the path loss model learned previously. If significant changes occur, the path loss model and GNN can be retrained periodically or as needed.

The upper section of Fig. 3 shows the training and prediction process for the sensor-centric GNN. The GNN is trained using



(a) REM of RSS residues of 15 sensors and 1 transmitter (b) Correlation matrix of real and predicted RSS residues

Fig. 4: GNN prediction results for new sensors.



(a) REM of RSS residues for 16 transmitters and 1 sensor (b) Correlation matrix of real and predicted RSS residues

Fig. 5: GNN prediction results for new transmitters.

data from five crowdsourced sensors, labeled Rx1, Rx4, Rx5, Rx6, and Rx8, along with two transmitters, labeled Tx1 and Tx2. Each transmitter generates a time sequence of RSS measurements at each of the five sensors, creating a sample that includes five time series of RSS residues. In this example, sensors Rx5 and Rx6 are masked in each sample. During system operation, when a new sensor Rx2 is introduced, the GNN can predict its RSS residues for both Tx1 and Tx2. In practice, a larger number of sensors and transmitters are used for training.

Fig. 6 presents exemplary results of GNN-based RSS residue prediction for both unmasked and masked sensors. Specifically, Fig. 4a shows Radio Environment Maps (REMs) illustrating the predicted RSS residues for 15 testing sensors. The GNN model successfully reconstructs the RSS residues for these sensors, demonstrating strong spatial correlations among them. Additionally, Fig. 4b demonstrates the correlation between matrices of actual and predicted RSS residues. Each column in these matrices represents residue values across all sensors for a specific transmitter (with 28 transmitters in total). The correlation matrix reveals the linear correlation between actual and predicted residue values from both matrices. Strong correlations are observed primarily between residue values from the same transmitter, confirming the distinct spatial RSS residue correlations among sensors associated with each transmitter.

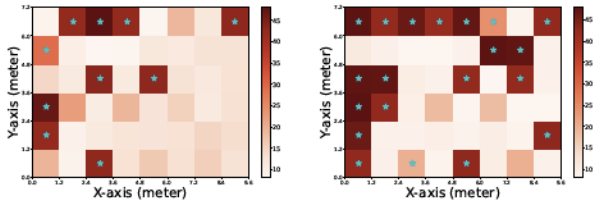
GNN Training for Unseen Transmitters. The second step involves training an inductive transmitter-centric GNN model specifically to predict the RSS residues generated by new transmitters that are not seen during the training phase—these are the transmitters that need to be localized. As described in Eq. (6) and illustrated before, strong spatial correlations are observed among the RSS residues from different transmitters at the same sensor. Therefore, we are motivated to employ a spatial interpolation method using the GNN model trained with data from known transmitters. By leveraging the GNN model's ability to understand spatial correlations, our approach can deliver reliable and accurate predictions of RSS residue values during system runtime, even for previously unseen transmitters, such as new spectrum misusers.

We use a grid-based method to generate training data. Specifically, the SAS operator creates an M-by-M grid over its sensing area, resulting in M^2 equally sized cells. We assume the SAS operator can either identify or dispatch a transmitter near each cell center. Each of these M^2 transmitters takes turns transmitting for a brief period, with only one being active at any given time. Each available sensor in V acts as a receiver, collecting a series of L RSS measurements. Using these measurements, the SAS operator calculates L RSS residues for each transmitter, as described by Eq. (3). Each training sample is linked to a specific sensor in V and is represented as a $M^2 \times L$ matrix, where each row corresponds to the L RSS residues for the transmitter located at the respective cell center. The SAS operator can generate as many samples as required for each transmitter, and this sample generation process can be integrated with the path loss model learning.

Given these RSS residue samples, we apply the same random-masking strategy and use the model structure outlined in Eq. (7) to train a three-layer transmitter-centric inductive GNN. The details of training and prediction processes are omitted here to avoid redundancy. The lower section of Fig. 3 illustrates the training and prediction process for this transmitter-centric GNN. The GNN is trained using data from five transmitters, labeled Tx1, Tx2, Tx3, Tx4, and Tx5, and three receivers, labeled Rx1, Rx2, and Rx3. Each receiver receives a time sequence of RSS measurements from each of the five transmitters, resulting in a sample that includes five time series of RSS residues. In this example, transmitters Tx5 and Tx6 are masked in each sample. During system operation, when a new transmitter Tx7 is introduced, the GNN can predict its RSS residues for all three receivers. In practice, a larger number of sensors and transmitters are used for training.

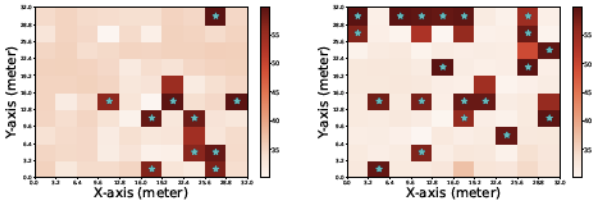
Fig. 5 shows some exemplary results of GNN-based RSS residue prediction for both unmasked and masked transmitters. Fig. 5a provides the REMs illustrating the predicted RSS residues for 16 testing transmitters with a single sensor. The GNN model successfully reconstructs the RSS residues for these transmitters, highlighting strong spatial correlations among them. Fig. 5b presents the correlation matrix of the real and predicted RSS residues for 10 testing sensors, demonstrating the accuracy of the GNN model in predicting RSS residues for new transmitters during system runtime.

The granularity of the grid partitioning, or equivalently the number M^2 of cells, significantly impacts localization precision. A finer grid with sufficient training transmitters results in more accurate RSS residue estimations during the online SML phase, due to the abundance of training samples



(a) 10 transmitters (b) 20 transmitters

Fig. 6: Visualizations of transmission power estimation in indoor lab environment with $P_0 = 45$dB.



(a) 10 transmitters (b) 20 transmitters

Fig. 7: Visualizations of transmission power estimation in outdoor parking-lot environment with $P_0 = 58$dB.

from various transmitters. Conversely, a coarser grid with fewer training transmitters may lead to poor generalization and less accurate RSS residue estimations. Therefore, selecting an appropriate number of training transmitters and grid cells is crucial for precise transmitter localization. In this paper, we use a cell size of $1.2\,\text{m} \times 1.2\,\text{m}$ for the $9.6\,\text{m} \times 7.2\,\text{m}$ indoor testbed with 48 transmitters during training, and the same cell size for the $32\,\text{m} \times 32\,\text{m}$ parking-lot testbed with 100 transmitters. To ensure consistency and accuracy, we apply the same grid partitioning during system runtime, which helps maintain reliable and precise RSS residue estimations for potential transmitters in the online SML phase.

C. GNN-based SML with RSS Residues and Zero-Forcing

Finally, we demonstrate how to combine receiver-centric and transmitter-centric GNNs in the SML phase to predict RSS residues for both sensors and transmitters that are not seen during training. We also illustrate how to integrate zero-forcing with these predicted RSS residues to achieve simultaneous, accurate, and rapid SML. Recall that V and U denote all the sensors available to the SAS operator and the transmitters to be localized during the SML phase. In our method, we assume that there could be a candidate transmitter at each of the M^2 cell centers. Thus, U initially includes all these M^2 candidate transmitters. Our technique is to identify the actual transmitters among them based on their approximate locations at the corresponding cell centers. For clarity, we refer to the sensors and transmitters used during training as historical sensors and transmitters, and those in V and U as new sensors and transmitters, respectively. Note that there may be overlap between historical and new sensors and transmitters, which can be easily managed with a simple modification of the process outlined below.

Step 1. We input all new sensors in V along with the RSS residue samples from the training phase into the receiver-centric GNN to predict RSS residues for each new sensor and historical transmitter. Because the GNN is location-centric and sensor-agnostic, all RSS residues—whether newly predicted or previously obtained during training—are associated with their respective (sensor, transmitter) locations. As shown in the upper-right section of Fig. 3, this process effectively expands the sample for each historical transmitter to include the new sensors.

Step 2. We then feed all historical and new RSS residue samples into the transmitter-centric GNN to predict RSS residues for each new sensor in V and candidate transmitter in U. As shown in the lower-right section of Fig. 3, this process effectively expands the sample for each new sensor in V to include all the candidate transmitters.

Step 3. With the predicted RSS residue $\mathrm{s}_{v\leftarrow u}$ for each sensor $v \in V$ and candidate transmitter $u \in U$, and the learned path loss model parameters (P_0, α), we use the results in Eq. (4) to compute the channel gain

$$h_{v\leftarrow u} = 10^{-P_0/10} \left(\frac{d_{v\leftarrow u}}{d_0}\right)^{-\alpha} e^{\tilde{\mathrm{s}}_{v\leftarrow u}},$$

where $\tilde{\mathrm{s}}_{v\leftarrow u} = \frac{\ln 10}{10}\mathrm{s}_{v\leftarrow u}$.

Next, we utilize the channel gain information and zero-forcing [15] to separate the superimposed RSS measurements $\{\mathbf{x}_v\}_{v\in V}$ for all M^2 candidate transmitters in U. Given $|\mathbf{V}|$ sensors and M^2 candidate transmitters, we define a $|\mathbf{V}| \times M^2$ matrix $\mathbf{H}$, where each element represents the channel gain between one sensor and one transmitter. Specifically, $\mathbf{H} = [\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_{M^2}]$, where $\vec{h}_u = [h_{v_1\leftarrow u}, h_{v_2\leftarrow u}, \ldots, h_{v_{|\mathbf{V}|}\leftarrow u}]^\intercal$ denotes the channel gain vector from transmitter $u \in U$ to all sensors in V. Zero-forcing [15] aims to find a decoding vector $\vec{C}_u$ for each transmitter u such that the signal from each transmitter can be decoupled from the superimposed RSS measurements $\{\mathbf{x}_v\}_{v\in V}$. Based on linear least squares (LLS) approximation theory [32], the decoding matrix $\{\vec{C}_1, \vec{C}_2, \ldots, \vec{C}_{M^2}\}$ is derived as the pseudo-inverse of the channel matrix:

$$[\vec{C}_1^\intercal, \vec{C}_2^\intercal, \ldots, \vec{C}_{M^2}^\intercal]^\intercal = (\mathbf{H}^\intercal\mathbf{H})^{-1}\mathbf{H}^\intercal. \tag{8}$$

With the decoding matrix, the transmission power vector $\mathbf{P} = [P_{u_1}, P_{u_2}, \ldots, P_{u_{M^2}}]$ of all candidate transmitters can be obtained by

$$\mathbf{P} = [\vec{C}_1^\intercal, \vec{C}_2^\intercal, \ldots, \vec{C}_{M^2}^\intercal]^T\mathbf{X} = (\mathbf{H}^\intercal\mathbf{H})^{-1}\mathbf{H}^\intercal\mathbf{X}, \tag{9}$$

where $\mathbf{X} = [\mathbf{x}_{v_1}, \mathbf{x}_{v_2}, \ldots, \mathbf{x}_{v_{|\mathbf{V}|}}]^\intercal$ represents the received RSS vector of all sensors in V.

Step 4. A transmitter is considered at the center of a specific cell if its power estimate exceeds a predefined system threshold, as shown in Fig. 6 and 7, which could be near the FCC power limit. If this cell-center location is not associated with an active spectrum authorization that minimally specifies the approved channel, location, duration, and power limit, the SAS operator classifies the transmitter as a spectrum misuser. In such cases, the operator may take actions, including involving law enforcement to physically exclude the spectrum misuser.

Furthermore, even if an active spectrum authorization is associated with the transmitter's location, the SAS operator may use crowdsourced spectrum misuser detection mechanisms, such as those described in [33], [34], [35], to ascertain whether the transmitter is a legitimate user or a spectrum misuser opportunistically operating within an authorized area.

IV. Performance Evaluation

A. Evaluation Settings

Our evaluations utilize the following two real datasets from [2] to facilitate the comparison between GNN-SML and related work. Due to space constraints, we leave the presentation of similar results obtained from large-scale simulations to a future extension of this work.

Dataset A: indoor testbed: The dataset [2] was collected in an indoor lab with dimensions of 9.6 m $\times$ 7.2 m, subdivided into 48 grid cells, each measuring 1.2 m $\times$ 1.2 m. During data collection, a single transmitter was moved through each cell, while each sensor, randomly placed in one of the grid cells, records average RSS readings. This setup results in a total of $48 \times 18 = 864$ RSS measurements.

Dataset B: outdoor testbed: This dataset [2] was collected in an open-space parking lot with dimensions of 32 m $\times$ 32 m, divided into 100 grid cells, each covering an area of 3.2 m $\times$ 3.2 m. Data collection involved moving a single transmitter through each cell, while 18 sensors, each randomly placed in one of the grid cells, record average RSS readings. In total, there are 1,800 RSS measurements.

We continue using V and U to represent the sensor and transmitter sets, respectively. For dataset A, we set the numbers of training and testing transmitters as $|U_{\text{train}}| = 28$ and $|U_{\text{test}}| = 20$, respectively, while for dataset B, we use $|U_{\text{train}}| = 50$ and $|U_{\text{test}}| = 20$, unless specified otherwise. Additionally, we use the same numbers of training and testing sensors as $|V_{\text{train}}| = |V_{\text{test}}| = 18$ for both datasets. To ensure statistical significance, we repeat each experiment 20 times by randomly selecting spectrum misusers from the transmitter set, and we report the average results below. We use three main performance metrics to evaluate and compare GNN-SML with state-of-the-art methods: transmitter localization error, transmission power estimation error, and localization time. Note that if each transmitter is considered a spectrum misuser, the localization and power estimation errors for transmitters are equivalent to those for spectrum misusers. Therefore, we do not distinguish between these terms in our evaluations.

B. Transmitter Localization Error

We measure the transmitter localization error using the Optimal SubPattern Assignment (OSPA) metric [36], [3]. The OSPA metric is a distance measure between two point patterns that accounts for both the dissimilarity in the number of points in the sets and the track values of these points. The localization error e_l is defined as

$$e_l(\theta, \hat{\theta}) := \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d(\theta_i, \hat{\theta}_{\pi(i)})^2 + c^2(n-m) \right) \right)^{\frac{1}{2}} \tag{10}$$

where θ and $\hat{\theta}$ are the ground-truth and estimated transmitter sets, with sizes m and n, respectively. Π_n denotes the set of all permutations of $\{1, \ldots, n\}$. The function $d(\theta_i, \hat{\theta}_{\pi(i)})$ represents the Euclidean distance between θ_i and $\hat{\theta}_{\pi(i)}$, with c as a cut-off constant (set to $c = 1$ in our case). Note that Eq. (10) applies when $m \leq n$; for $m > n$, we use $e_l(\theta, \hat{\theta}) := e_l(\hat{\theta}, \theta)$. Thus, Eq. (10) captures both the average distance estimation error between the ground-truth and estimated transmitter sets and the error due to differences in their sizes.

We compare the transmitter localization error of GNN-SML with state-of-the-art methods, including SPLOT [3], MAP* [2], LLOCUS (with the RRBI mode) [4], and the classical kNN method [4]. For both kNN and LLOCUS, we use the log-distance path loss model in Eq. (4) for transmission power estimation to ensure a fair comparison.

Fig. 8a and Fig. 8b illustrate the localization errors for these schemes across both datasets as the number of transmitters (i.e., spectrum misusers) increases from 1 to nearly half of the grid cells. The results highlight GNN-SML's significant advantages over previous methods. Specifically, while SPLOT shows the lowest localization error among the four comparison methods, GNN-SML outperforms it by 37% and 58% for dataset A (indoor) and dataset B (outdoor), respectively, with 20 misusers. This improvement is due to kNN and LLOCUS assuming that RSS measurements are primarily influenced by a single transmitter, which increases noise and errors with more misusers. GNN-SML's superior performance is attributed to its ability to handle spatial and temporal variability, whereas SPLOT relies on fixed parameters and overlooks these factors, leading to less accurate localization.

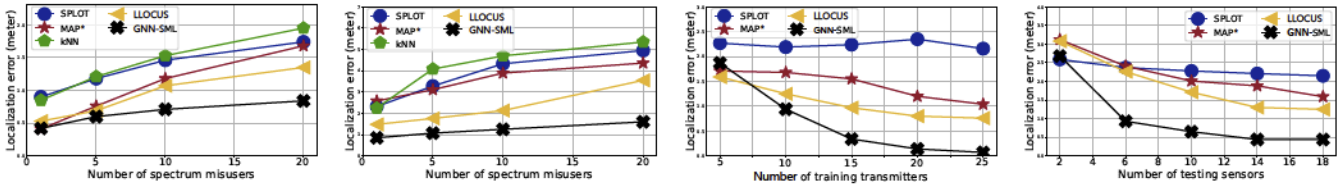
We also evaluate how GNN-SML's localization error varies with different numbers of training transmitters and testing sensors, i.e., $|U_{\text{train}}|$ and $|V_{\text{test}}|$. Fig. 8c shows that as $|U_{\text{train}}|$ increases, GNN-SML, LLOCUS, and MAP* achieve significantly lower localization errors because more training data improves the quality of their respective ML models. Similarly, Fig. 8d demonstrates that the localization errors of GNN-SML, LLOCUS, and MAP* decrease as $|V_{\text{test}}|$ increases, since more sensors involved in the SML phase lead to better prediction results. In contrast, SPLOT, being a non-learning-based method, remains relatively unaffected by changes in $|U_{\text{train}}|$ and $|V_{\text{test}}|$. Nonetheless, GNN-SML consistently outperforms all other methods in every evaluated scenario.

C. Transmission Power Estimation Error

Similar to the OSPA-based localization error metric, we define the transmission power estimation error e_p as

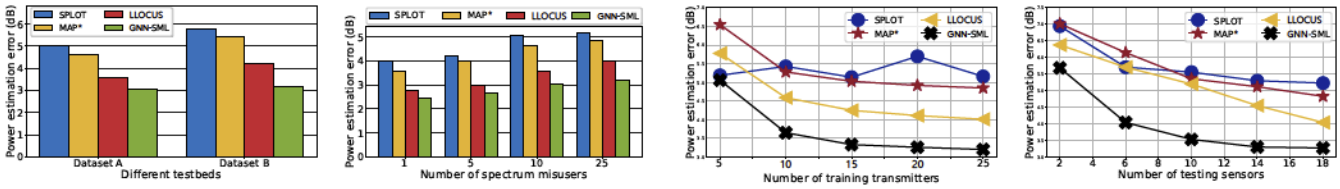
$$e_p(\theta, \hat{\theta}) := \left(\frac{1}{n} \left(\sum_{i=1}^{m} \left| P_{\theta_i} - P_{\hat{\theta}_{\pi(i)}} \right|^2 \right) \right)^{\frac{1}{2}}, \tag{11}$$

where P_{θ_i} and $P_{\hat{\theta}_{\pi(i)}}$ represent the real and estimated transmission powers corresponding to the ground-truth and estimated transmitter sets θ and $\hat{\theta}$, respectively. It is clear that e_p can be influenced by both the transmitter cardinality error $|m-n|$ and



(a) Dataset A (b) Dataset B (c) Dataset A (d) Dataset A

Fig. 8: Transmitter localization errors in indoor and outdoor environments.



(a) Dataset A and B (b) Dataset A (c) Dataset A (d) Dataset A

Fig. 9: Transmission power estimation errors in indoor and outdoor environments.

the differences between the real and estimated transmission powers.

Fig. 9a and Fig. 9b demonstrate the significant improvement of GNN-SML over SPLOT, MAP*, and LLOCUS. For clarity, the poorest results using kNN are not shown for both indoor and outdoor environments. Specifically, as shown in Fig. 9a, while LLOCUS achieves lower power estimation errors than SPLOT and MAP*, GNN-SML still outperforms it by 14% and 24% for dataset A (indoor) and dataset B (outdoor), respectively, when there are $|U_{\text{test}}| = 20$ spectrum misusers in both datasets. Additionally, Fig. 9b reveals that as $|U_{\text{test}}|$ increases, the power estimation errors also increase for all schemes due to more spectrum misusers causing closer average distances among them and greater difficulty in disentangling their transmissions at the sensors. However, GNN-SML consistently shows a clear advantage over all other schemes.

Fig. 9c and Fig. 9d illustrate how the power estimation errors of all schemes change with $|U_{\text{train}}|$ and $|V_{\text{test}}|$. Similar to localization errors, the power estimation errors for GNN-SML, LLOCUS, and MAP* decrease dramatically as $|U_{\text{train}}|$ or $|V_{\text{test}}|$ increases. This is because more transmitters involved in the training phase and more sensors in the SML phase improve the prediction accuracy of their respective ML models. In contrast, SPLOT remains relatively unaffected by $|U_{\text{train}}|$ and $|V_{\text{test}}|$ due to its lack of an ML component. Nonetheless, GNN-SML demonstrates better performance in all cases. For example, as shown in Fig. 9c, GNN-SML outperforms LLOCUS and MAP* by approximately 20% and 34%, respectively, when 25 transmitters are used during training for dataset A.

D. Localization Time (Prediction Time)

We also evaluate the localization time (i.e., online prediction time) of GNN-SML. The training and testing are conducted on a Dell desktop with a 2.90 GHz CPU, 32 GB RAM, and Windows 10 64-bit Professional. Our experiments show that GNN-SML achieves an average localization time of less than 1.2 s, which is slightly longer than LLOCUS's localization time of 1.1 s [4]. While GNN-SML employs deep learning techniques, LLOCUS uses traditional machine learning methods, making their localization times comparable. However, GNN-SML significantly outperforms other deep learning-based SML methods [9], [37]. For instance, DeepTxFinder [9] has a prediction complexity of $\mathcal{O}(|N||M|^2)$, where N and M represent the number of sensors and transmitters, respectively, while GNN-SML has a lower complexity of $\mathcal{O}(|M|^2)$. Additionally, DeepMTL [37] incurs even longer running times due to its deep YOLOv3-cust component, which consumes over 90% of the total runtime.

In summary, GNN-SML offers comparable computational performance to the most efficient method available (i.e., LLOCUS [4]) while providing significantly improved localization and power estimation accuracy. Thus, GNN-SML is highly suitable for assisting SAS operators in identifying spectrum misusers in nearly real-time.

V. Conclusion

We introduce GNN-SML, a novel framework for accurate and fast localization of spectrum misusers in crowdsourced DSS systems. Leveraging location-centric, sensor-agnostic GNNs with inductive capabilities, it can simultaneously localize multiple misusers, even in close proximity with unknown locations and transmission powers. Validation on real datasets shows GNN-SML outperforms existing methods.

Acknowledgment

This work was supported in part by the U.S. National Science Foundation under grant CNS-2325563 and CNS-2422863. Research was also sponsored in part by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-23-2-0225. Research was also supported in part by the STARs program of the University of Texas System under grant AR911847. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the University of Texas System, the Army Research Laboratory, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] A. Bhattacharya, C. Zhan, H. Gupta, S. Das, and P. Djuric, "Selection of sensors for efficient transmitter localization," in *INFOCOM*, Las Vegas, Nevada, October 2020.

[2] C. Zhan, H. Gupta, A. Bhattacharya, and M. Ghaderibaneh, "Efficient localization of multiple intruders in shared spectrum system," in *IPSN*, San Jose, CA, May 2020.

[3] 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 *MOBICOM*, Vienna, Austria, June 2017.

[4] S. Sarkar, A. Baset, H. Singh, P. Smith, N. Patwari, S. Kasera, K. Derr, and S. Ramirez, "LLOCUS: learning-based localization using crowdsourcing," in *MOBIHOC*, Scotland, United Kingdom, October 2020.

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

[6] A. Nika, Z. Li, Y. Zhu, Y. Zhu, B. Zhao, X. Zhou, and H. Zheng, "Empirical validation of commodity spectrum monitoring," in *ACM SenSys*, Standard, CA, November.

[7] R. Zhang, J. Zhang, Y. Zhang, and C. Zhang, "Secure crowdsourcing-based cooperative spectrum sensing," in *IEEE INFOCOM*, Turin, Italy, April 2013.

[8] X. Jin and Y. Zhang, "Privacy-preserving crowdsourced spectrum sensing," in *IEEE INFOCOM*, San Francisco, CA, April 2016.

[9] A. Zubow, S. Bayhan, P. Gawłowicz, and F. Dressler, "DeepTxFinder: Multiple transmitter localization by deep learning in crowdsourced spectrum sensing," in *IEEE ICCCN*, Honolulu, Hawaii, USA, August 2020.

[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, pp. 1574–1192, June 2022.

[11] C. Ezemaduka and A. Abouzeid, "Privacy-aware deep learning based localization of spectrum violators," in *2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2023, pp. 98–106.

[12] C. Zhan and H. Gupta, "Quantum sensor network algorithms for transmitter localization," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1. IEEE, 2023, pp. 659–669.

[13] I. Bizon, A. Nimr, P. Schulz, M. Chafii, and P. Fettweis, "Blind transmitter localization using deep learning: a scalability study," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2023, pp. 1–6.

[14] F. Scarselli, M. Gori, A. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, January 2009.

[15] Q. Spencer, A. L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser mimo channels," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 461–471, 2004.

[16] Y. Hu and R. Zhang, "A spatiotemporal approach for secure crowdsourced radio environment map construction," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1790–1803, 2020.

[17] Y. Zhang, A. Li, J. Li, D. Han, T. Li, R. Zhang, and Y. Zhang, "SpecKriging: Gnn-based secure cooperative spectrum sensing," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 9936–9946, November 2022.

[18] X. Jin, R. Zhang, Y. Chen, T. Li, and Y. Zhang, "Dpsense: Differentially private crowdsourced spectrum sensing," in *ACM CCS*, Vienna, Austria, October 2016.

[19] W. C. Lee, *Mobile communications engineering: theory and applications*. McGraw-Hill, Inc., 1997.

[20] J. Riihijarvi, P. Mahonen, M. Wellens, and M. Gordziel, "Characterization and modelling of spectrum for dynamic spectrum access with spatial statistics and random fields," in *IEEE PIMRC*, 2008.

[21] M. Wellens, J. Riihijärvi, and P. Mähönen, "Spatial statistics and models of spectrum use," *Computer Communications*, vol. 32, no. 18, pp. 1998–2011, 2009.

[22] R. Olea, "A six-step practical approach to semivariogram modeling," *Stochastic Environmental Research and Risk Assessment*, vol. 20, no. 5, pp. 307–318, 2006.

[23] H. Alaya-Feki, S. Jemaa, B. Sayrac, P. Houze, and E. Moulines, "Informed spectrum usage in cognitive radio networks: Interference cartography," in *IEEE PIMRC*, Cannes, France, August 2008.

[24] A. Achtzehn, J. Riihijärvi, G. Vargas, M. Petrova, and P. Mähönen, "Improving coverage prediction for primary multi-transmitter networks operating in the tv whitespaces," in *IEEE SECON*, Seoul, Korea, June 2012.

[25] X. Ying, C. Kim, and S. Roy, "Revisiting tv coverage estimation with measurement-based statistical interpolation," in *COMSNETS*, Bangalore, India, January 2015.

[26] X. Ying, S. Roy, and R. Poovendran, "Incentivizing crowdsourcing for radio environment mapping with statistical interpolation," in *IEEE DySPAN*, Stockholm, Sweden, September 2015.

[27] A. Konak, "A kriging approach to predicting coverage in wireless networks," *International Journal of Mobile Network Design and Innovation*, vol. 3, no. 2, pp. 65–71, 2009.

[28] H. Braham, S. Jemaa, B. Sayrac, G. Fort, and E. Moulines, "Low complexity spatial interpolation for cellular coverage analysis," in *WiOpt*, Hammamet, Tunisia, May 2014.

[29] Y. Hu and R. Zhang, "A spatiotemporal approach for secure crowdsourced radio environment map construction," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1790–1803, 2020.

[30] C. Phillips, M. Ton, D. Sicker, and D. Grunwald, "Controlling and monitoring process in industrial automation using zigbee," in *DySPAN*, Bellevue, WA, October 2012.

[31] Y. Wu, D. Zhuang, A. Labbe, and L. Sun, "Inductive graph neural networks for spatiotemporal kriging," in *AAAI*, Virtual Conference, February 2021.

[32] G. Watson, "Linear least squares regression," *The Annals of Mathematical Statistics*, pp. 1679–1699, 1967.

[33] L. Yang, Z. Zhang, B. Zhao, C. Kruegel, and H. Zheng, "Enforcing dynamic spectrum access with spectrum permits," in *ACM MobiHoc*, Hilton Head Island, SC, June 2012.

[34] X. Jin, J. Sun, R. Zhang, and Y. Zhang, "SafeDSA: Safeguard dynamic spectrum access against fake secondary users," in *ACM CCS*, Denver, CO, October 2015.

[35] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "Specguard: Spectrum misuse detection in dynamic spectrum access systems," in *IEEE INFOCOM*, Hong Kong, China, April 2015.

[36] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[37] C. Zhan, M. Ghaderibaneh, P. Sahu, and H. Gupta, "DeepMTL: Deep learning based multiple transmitter localization," in *IEEE WoWMoM*, Paris, France, September 2021.