OrchLoc: In-Orchard Localization via a Single LoRa Gateway and Generative Diffusion Model-based Fingerprinting

Kang Yang
University of California Merced
Merced, USA
kyang73@ucmerced.edu

Yuning Chen
University of California Merced
Merced, USA
ychen372@ucmerced.edu

Wan Du
University of California Merced
Merced, USA
wdu3@ucmerced.edu

ABSTRACT

In orchards, tree-level localization of robots is critical for smart agriculture applications like precision disease management and targeted nutrient dispensing. However, prior solutions cannot provide adequate accuracy. We develop our system, a fingerprinting-based localization system that can provide tree-level accuracy with only one LoRa gateway. We extract channel state information (CSI) measured over eight channels as the fingerprint. To avoid labor-intensive site surveys for building and updating the fingerprint database, we design a CSI Generative Model (CGM) that learns the relationship between CSIs and their corresponding locations. The CGM is fine-tuned using CSIs from static LoRa sensor nodes to build and update the fingerprint database. Extensive experiments in two orchards validate our system's effectiveness in achieving tree-level localization with minimal overhead and enhancing robot navigation accuracy.

CCS CONCEPTS

• **Networks** → **Location based services**; *Wireless access points, base stations and infrastructure*; • **Computer systems organization** → *Sensor networks*.

KEYWORDS

In-Orchard Localization, LoRaWAN, Fingerprinting, Generative Diffusion Model

ACM Reference Format:
Kang Yang, Yuning Chen, and Wan Du. 2024. *OrchLoc*: In-Orchard Localization via a Single LoRa Gateway and Generative Diffusion Model-based Fingerprinting. In *The 22nd Annual International Conference on Mobile Systems, Applications and Services (MOBISYS '24), June 3–7, 2024, Minato-ku, Tokyo, Japan.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3643832.3661876

1 INTRODUCTION

In modern orchards, robots have become indispensable for executing precision agriculture practices, such as pruning, harvesting, and spraying [1, 2]. These robots are tasked with focusing on individual trees for specific operations. For instance, conducting proactive health assessments on individual trees facilitates timely interventions that can prevent the spread of diseases [3]. To perform these functions, robots require the capability to accurately identify each tree and navigate within orchards. Considering the spacing between two adjacent trees is typically 4.9 m [4], achieving tree-level identification necessitates localization accuracy finer than 4.9 m.

While tree-level localization is readily achievable in urban environments [5, 6], achieving tree-level localization in orchards presents significant challenges. Traditional methods for robot localization, such as wheel encoders [7], SLAM (Simultaneous Localization and Mapping)[8], and GPS (Global Positioning System)/INS (Inertial Navigation System)[9], fall short in accurately identifying individual trees. Specifically, wheel encoders are prone to slipping in muddy conditions, compromising their reliability [10]. The SLAM systems face difficulties with variable lighting conditions, obstructions, high infrastructure costs, high power requirements, and uneven terrain [11, 12]. The GPS/INS systems, known for their robustness, low cost, and energy efficiency, rely on GPS to provide precise positioning that compensates for the inherent drift in inertial sensors [9, 13]. However, GPS accuracy is notably compromised in orchards due to signal obstruction by tree canopies [9, 14]. Our experiments conducted in a pistachio orchard demonstrated an average GPS localization error of 7.9 m, which led to a navigation accuracy of 9.1 m using the state-of-the-art GPS/INS algorithm [9]. Consequently, such level of accuracy renders traditional robot navigation methods impractical in orchard environments.

Recent advancements in precision agriculture have seen the deployment of long-range, low-power LoRa networks [15, 16] in orchards for many applications such as smart irrigation [17, 18] and pest monitoring [19, 20]. This work explores the innovative use of existing LoRa infrastructure to enable robot localization in orchards. By equipping a robot with a LoRa node, it transmits packets to a nearby LoRa gateway. This gateway then estimates the robot's location based on these packets and relays the calculated position back to the robot. Moreover, by integrating LoRa-based localization with inertial navigation systems, we can improve navigation accuracy.

Several algorithms for LoRa localization have been proposed [21–24]. However, their application in orchards faces challenges: 1) These algorithms require the reception of the same packet by multiple gateways for effective localization. In contrast to urban environments, LoRa gateways in orchards are deployed more sparsely, aimed at covering large areas at minimal costs. Consequently, at most locations, the signal from a sensor node is likely to be received by only a single gateway. 2) They assume Line-of-sight (LoS) signal paths between nodes and gateways. In orchards, however, gateways are mounted on high poles for broad coverage and nodes are installed under canopies for environmental measurements. LoS paths barely exist due to the blockage of tree canopies.

MOBISYS '24, June 3–7, 2024, Minato-ku, Tokyo, Japan

ACM ISBN 979-8-4007-0581-6/24/06
https://doi.org/10.1145/3643832.3661876

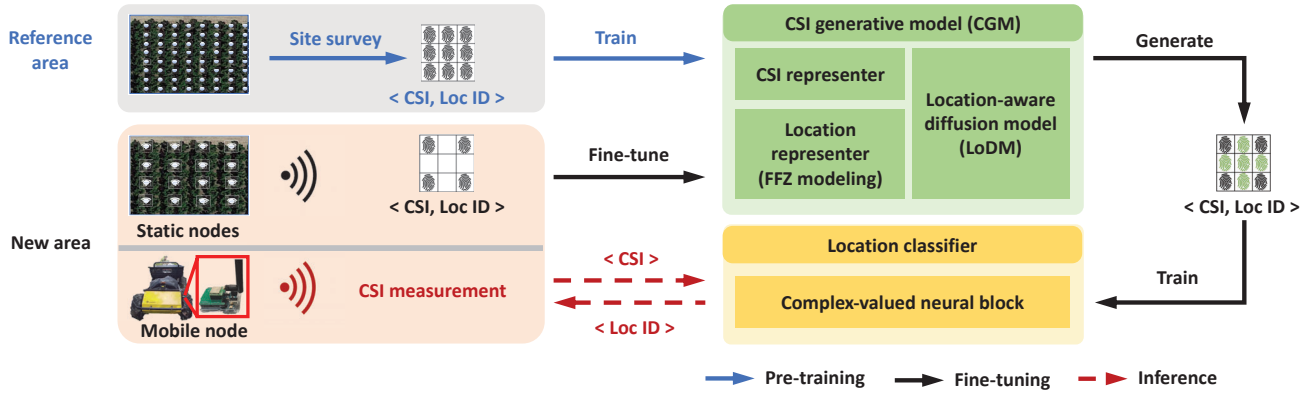


Figure 1: The workflow of *OrchLoc*. The combination of pre-training and fine-tuning stages is termed the turbo-training scheme.

This paper introduces *OrchLoc*, a novel fingerprinting-based LoRa localization system in orchards using a single LoRa gateway. We propose a new fingerprint for LoRa localization, *i.e.*, Channel State Information (CSI), which is extracted from both the amplitude and phase spectrum of signals received across eight channels by a dual-antenna gateway. Distinct from Wi-Fi CSI fingerprinting methods that separately utilize amplitude [25] or phase [26, 27] for localization, we devise a location classifier with complex-valued Fully-Connected (FC) blocks to process CSI measurements. Experimental results reveal that our location classifier enhances amplitude-only and phase-only methods in *precision*, with improvements of 20.3% and 46.7%, respectively.

However, *OrchLoc* encounters two challenges: the labor-intensive site surveys in large orchards and the need for periodic updates to counteract fingerprint aging. This paper highlights two key observations enabling efficient database construction and updates with minimal overhead: *1) Media Homogeneity:* Existing sensor nodes deployed across orchards for precision agriculture are utilized to periodically provide CSI fingerprints from their stationary locations. The challenge arises in updating fingerprints for locations without sensor nodes. Although distinct CSI patterns emerge across various locations, LoRa signals only traverse through three media: air, foliage, and ground. Experiments show that shadowing effects from these media are consistent across all locations. This consistency means shadowing effects learned from locations with sensor nodes can be applied to refresh fingerprints at locations lacking sensor nodes. *2) Spatial Homogeneity:* Modern orchards feature uniformly shaped trees and systematic layouts, allowing for the even distribution of LoRa gateways to ensure comprehensive coverage. This uniformity permits the division of an orchard into distinct gateway coverage areas, each with a consistent tree layout. Each gateway is responsible for maintaining its own fingerprint database. Our experimental findings reveal a high similarity in CSI fingerprints across these various areas. This similarity supports the practicality of utilizing the database from one area to inform the databases of other areas.

Inspired by recent advancements in generative models [28], we introduce a *CSI Generative Model (CGM)* designed to synthesize CSI fingerprints based on location IDs. As depicted in Figure 1, the CGM is central to our proposed turbo-training scheme, which facilitates the construction and updating of fingerprint databases in orchards. Initially, the CGM is pre-trained with fingerprints from a reference area, selected for its *Spatial Homogeneity*—a characteristic ensuring the representativeness of other areas. Subsequently, this pre-trained CGM undergoes fine-tuning with data from sensor nodes deployed in a new target area. This fine-tuning phase is crucial for capturing local environmental features, significantly improving the model's adaptability to the specific conditions of the new area. We conceptualize the CGM as a "CSI propagation model", leveraging the *Media Homogeneity* to assert that the fine-tuned CGM can generate CSI data for locations devoid of sensor nodes. Through this process, we can continuously refresh the classifier with generated data, thereby circumventing the reliance on labor-intensive manual measurements for data collection.

Our CGM, featuring a location-aware diffusion model (LoDM), can generate CSI for each location. It contains a complex-valued U-Net framework [29] and attention layers [30, 31] to effectively learn the latent relationship between CSI data and location IDs. However, feeding raw CSI data and location IDs directly into the LoDM could constrain its modeling capabilities. This limitation arises because location IDs do not provide detailed location-specific information, and the utilization of low-dimensional CSI vectors could lead to overfitting. To mitigate this issue, the CGM integrates specialized CSI and location representers. A complex-valued auto-encoder-based CSI representer converts the data into high-dimensional vectors, effectively extracting latent features. The location representer computes the proportions of foliage, air, and ground within the First Fresnel Zone (FFZ). Combined with the communication distance and direction, our representer yields the FFZ vector. This vector offers a more detailed, physically-informed, and comprehensive representation of location compared to mere location IDs.

We collected CSI fingerprint database across seven rounds over four weeks in one area, and gathered CSI data in ten areas across two pistachio orchards. The experimental results show that our turbo-training scheme consistently maintained average *precision* and *recall* at 96.3% and 97.6%, respectively, over the four weeks. For the ten areas, *OrchLoc* achieved an average *precision* and *recall* of 89.6% and 91.8%, with a localization error of just 1.2 m. Furthermore, substituting GPS with *OrchLoc* for robotic navigation in orchards, employing the Neural-KF navigation algorithm [9], resulted in a reduction of navigation errors by 61.3%.

In summary, this paper makes the following contributions:

- We develop *OrchLoc*, an in-orchard fingerprinting-based localization system that achieves tree-level localization accuracy with a single LoRa gateway.
- We design CSI-based fingerprint and devise a complex-valued FC block as a classifier for location estimation.
- Based on two observations in orchards, a turbo-training scheme is developed, coupled with CGM, to enable efficient database construction and updating.
- To generate high-quality CSI fingerprints, we customize CGM by integrating a location-aware diffusion model, a CSI representer, and a FFZ-based location representer.
- Extensive experiments demonstrate the localization accuracy and utility for robot navigation in orchards.

2 MOTIVATION

This work aims to provide tree identification and robot navigation in orchards. A critical question is the impact of GPS accuracy on the navigation performance of GPS/INS systems in orchards. Utilizing the public dataset [9], we simulate a robot's movement across farm terrains, covering a total trajectory length of 2.0 km. The robot is equipped with a Bosch BNO055 INS unit [32] for capturing inertial data. Meanwhile, an OptiTrack 13W-P MoCa system [33] records the robot's initial positioning and continuous velocity to establish trajectory groundtruth. To mirror GPS accuracy in orchard scenarios—averaging 7.9 m as detailed in Section 7.4—we introduce Gaussian noise with a mean of 7.9 m and a standard deviation of 1.0 into the real coordinates. We test two methods that merge GPS and INS data to correct the drift of INS sensors, i.e., Extended Kalman Filter (EKF) [13] and Neural-KF [9].

Figure 2 reveals that navigation errors—measured as the average root-mean-square error (RMSE) between the predicted and actual location trajectories—decrease to 11.6 m with EKF and 9.1 m with Neural-KF. These substantial errors stem from the reliance on GPS for accurate position corrections to counteract INS sensor drift. In environments like orchards where GPS accuracy is significantly diminished, the system's capacity to amend INS drift is severely compromised, resulting in considerable navigation errors. Given that typical tree spacing of 4.9 m in orchards, these levels of error are impractical for autonomous robot navigation. This paper introduces an innovative approach that leverages an alternative to GPS for sensor drift correction, aiming to enhance localization accuracy in GPS-challenged environments.

3 LOCALIZATION USING *ORCHLOC*

We first introduce the CSI for LoRa signal, then devise a location classifier that integrates complex-valued FC blocks to process the complex-valued data, enabling CSI fingerprint-based localization.

3.1 CSI Fingerprint

3.1.1 Extracting CSI for LoRa. To extract stable CSI data for LoRa, we incorporate several operations: de-chirping, Fast Fourier Transform (FFT), preamble calibration, phase rotation compensation for amplitude calculation, and utilization of the phase difference between two antennas for phase estimation. Upon the reception

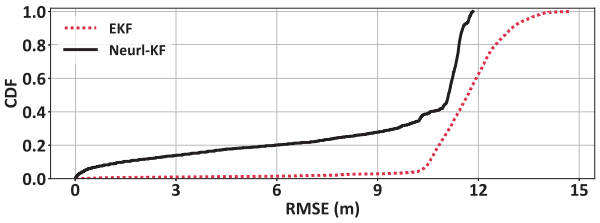


Figure 2: The robot navigation accuracy in orchards.

of a packet, the CSI (a complex number representing channel frequency response) is obtained by comparing the amplitude and phase spectrum of the packet preambles to those of a standard up-chirp. Preambles are well-calibrated as [34–36], serving as a reliable basis for the following calculation.

Amplitude: The process begins with de-chirping and applying FFT to the preambles, expressed as $Y(f) = \mathcal{F}[r(t)]$, where a peak is identifiable at the first frequency bin. The amplitude of CSI is determined by the ratio of the peak height of the received preamble to that of the standard up-chirp. To counteract the impact of FFT phase rotation on peak height estimation, phase rotation at frequencies f and $f - BW$ is compensated. This is achieved by optimizing the summation of $Y(f)\cdot e^{j\phi}$ and $Y(f-BW)$ over $\phi \in [0, 2\pi]$, facilitating an accurate estimation of the peak height.

Phase: We first calculate the phase of the peak from two antennas. Then, the phase difference between the two antennas is derived to obtain more stable phase data over time [34, 37].

These operations not only ensure the extraction of stable CSI data but also minimize the potential interference impact from other wireless systems operating within the same unlicensed frequency band. Notably, we do not account for cell edge interference from adjacent gateways, as in the US, adjacent LoRa gateways are configured to operate on different frequency channels [38]. This regulation prevents signal interference between neighboring gateways.

CSI Fingerprint $\mathbf{x}_0$: Unlike Wi-Fi, where a packet occupies multiple channels, a LoRa packet is sent on only one channel each time. Given a gateway operating across eight channels, nodes are instructed to transmit eight packets, with each on a separate channel. Thus, a fingerprint $\mathbf{x}_0$ is obtained with a dimension of 2×8, capturing the frequency response across eight channels for both antennas.

Database: CSI fingerprints are collected at all $M = 64$ locations within a gateway coverage area. At each location, 160 packets are collected. Each set of eight consecutive packets across the eight channels forms a CSI vector, yielding 20 CSI vectors per location. We also employ a common data augmentation technique [39] to produce 80 additional CSI vectors at each location by adding Gaussian noise to the initial 20 CSI vectors. This noise was sampled from a zero-mean Gaussian distribution. The standard deviation for each dimension was set according to the standard deviation calculated from the collected 20 CSI vectors.

3.1.2 Spatial Resolution. We employ the MUSIC algorithm [40] and t-SNE [41] to illustrate the spatial resolution of CSI from physical and representation learning perspectives.

The MUSIC algorithm is used on the LoRa signal samples from two antennas to assess the signal strength at various angles. Repeated across eight channels, this yields spatial spectra for each location.

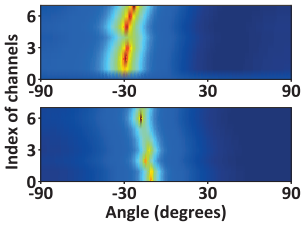


Figure 3: Spatial spectra of two adjacent locations in an area.

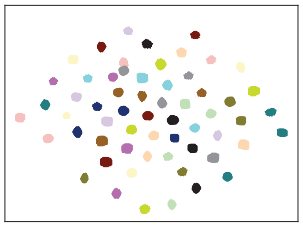


Figure 4: Representation of locations in an area using t-SNE.

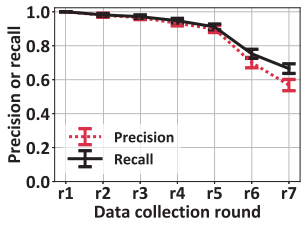


Figure 5: The accuracy over the seven rounds.

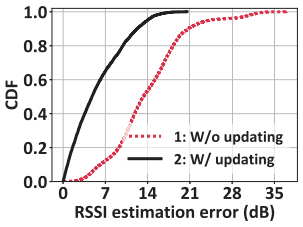


Figure 6: Updating by a subset of locations.

Figure 3 shows distinct spatial spectra for two neighboring locations, indicating different signal arrival angles at the gateway. This difference confirms the ability to distinguish two adjacent locations in orchards using LoRa signals.

Furthermore, Figure 4 presents the results obtained by applying t-SNE to the CSI fingerprints from all locations. Markers of different colors or shapes represent distinct locations. The CSI data from the same locations display a clustering effect, confirming the spatial resolution of CSI fingerprints.

3.2 Complex-Valued Location Classifier

It is intuitive to flatten the CSI vector into a one-dimensional real-valued vector. Then, traditional matching methods, such as the K-Nearest Neighbors (KNN), can be applied for location classification. However, the direct flattening might diminish the spatial or structural features inherent to the CSI vector, and compromise the inter-relations among vector elements. In contrast, we introduce the complex-valued FC block to effectively integrate both amplitude and phase information, thereby enhancing localization accuracy.

Complex-Valued FC Block: It consists of two real-valued neurons, each processing the real and imaginary parts of input, denoted by $\mathbf{u}$ and $\mathbf{v}$, respectively. If input data is represented as $\mathbf{c} = \mathbf{u} + i \cdot \mathbf{v}$, the weight matrix as $\mathbf{w} = \mathbf{w_u} + i \cdot \mathbf{w_v}$, and the bias as $\mathbf{b} = \mathbf{b_u} + i \cdot \mathbf{b_v}$, then the output of the block is given by $\mathbf{c}' = \mathbf{u}' + i \cdot \mathbf{v}'$. Here, $\mathbf{u}' = \sigma(\mathrm{Re}(\mathbf{w} \cdot \mathbf{c} + \mathbf{b}))$ and $\mathbf{v}' = \sigma(\mathrm{Im}(\mathbf{w} \cdot \mathbf{c} + \mathbf{b}))$, where σ is the nonlinear activation function such as *Rectified Linear Unit (ReLU)*.

Our classifier consists of two complex-valued FC blocks and two real-valued FC layers, each followed by a *ReLU* function. To bridge complex-valued FC blocks and FC layers, the absolute value of outputs from the second complex-valued block is computed as the input for the first FC layer. Features from the second FC layer are then directed into the third FC layer, equipped with M neurons and a *softmax* activation function.

3.3 Performance and Challenges

Over four weeks, we conducted seven rounds ($r1$ to $r7$) of CSI measurement in an area with $M = 64$ locations. In each round, we collect a fingerprint database for all locations, resulting in seven distinct databases corresponding to different times: $r1$ (first day AM), $r2$ (first day PM), $r3$ (second day), $r4$ (third day), $r5$ (tenth day), $r6$ (seventeenth day), and $r7$ (twenty-fourth day). The fingerprint measurement process needed around four hours in each round. The environmental parameters observed in each data collection round are documented in Table 1. The CSI fingerprint database from the first round ($r1$) was divided into training and testing sets in a $7:3$ ratio. The location classifier, trained solely with the $r1$ training set, underwent evaluation using the databases from rounds $r1$ to $r7$, notably without being retrained with databases from rounds $r2$ to $r7$.

Figure 5 illustrates the consistently high accuracy of our classifier across the first five data collection rounds ($r1 - r5$), with the average *precision* and *recall* stabilizing no lower than 89.8% and 91.1%, respectively. This stability is attributed to three principal factors: First, the integration of techniques that are proposed in Section 3.1.1, including de-chirping, FFT, phase rotation compensation for amplitude calculation, and the phase difference of two antennas for phase calculation, enhances the stability of CSI data [34, 37]. Second, the environmental conditions throughout this period were relatively stable, as evidenced by the data in Table 1. Specifically, the analysis of this environmental data reveals a average temperature increase of 3.1°C from round $r1$ to rounds $r2$ through $r5$, suggesting that the fingerprint database remains robust when temperature fluctuations are kept below this threshold. Moreover, orchards usually experience minimal external disturbances such as human activity or vehicular traffic, which further contributes to the stability observed.

Aging Problem: Figure 5 also presents a noticeable decline in the classifier's accuracy from round $r6$, with *precision* and *recall* dropping to 56.8% and 66.5%, respectively. This downturn is primarily ascribed to a significant shift in the distribution of fingerprints, prompted by the altered environmental dynamics at rounds $r6$ and $r7$ compared to those in round $r1$. In particular, Table 1 highlights a substantial temperature increase of 10.5°C from round $r1$ to rounds $r6$ and $r7$. This variance highlights the need for periodic refreshing of CSI fingerprints when temperature changes surpass the 3.1°C mark, underlining the importance of regular updates to the fingerprint database to preserve the accuracy of the classifier.

Vast Orchard: In a 100-acre orchard with row and column spacing of 6.7 m and 4.9 m, and LoRa communication range of 120 m, the orchard is divided into 29 areas, each with 19×26 trees (locations) [4]. Allocating three minutes per location for data collection, the total survey time amounts to 29.8 days. The requirement for database refreshment further complicates the process, rendering such site surveys infeasible.

3.4 Key Observations

We pinpoint two key observations in orchards that aid in reducing labor for building and updating database.

3.4.1 Media Homogeneity. A experiment focusing on Received Signal Strength Indicator (RSSI) estimation was conducted to explain this concept. The Log-Normal Shadowing model (Log) [4, 42] is employed to estimate the RSSI of the received signals. The model uses Path Loss Exponent (PLE) to represent environmental impacts

Table 1: Statistics of environmental dynamics observed across seven data collection rounds over a four-week period.

Date	Humidity (%)	Temp (°C)	Wind (m/s)
1st day AM (r1)	39.0 ± 8.5	26.7 ± 2.6	2.1 ± 0.1
1st day PM (r2)	34.3 ± 3.5	31.0 ± 0.9	1.9 ± 0.3
2nd day PM (r3)	42.3 ± 3.1	29.9 ± 0.4	2.1 ± 0.4
3rd day PM (r4)	43.7 ± 1.5	27.8 ± 1.5	2.1 ± 0.3
10th day PM (r5)	56.3 ± 18.6	31.2 ± 5.8	0.8 ± 0.4
17th day PM (r6)	39.3 ± 11.5	36.8 ± 2.7	1.0 ± 0.2
24th day PM (r7)	24.7 ± 3.5	35.5 ± 0.7	2.0 ± 0.0

on signal. In orchards, signals traverse three media: air, foliage, and ground. Therefore, PLE is decomposed into three components [4]:

$$PLE = P_{air} \times \alpha + P_{foliage} \times \beta + P_{ground} \times \gamma \quad (1)$$

where P_{air}, $P_{foliage}$, and P_{ground} are proportions of air, foliage, and ground among signal transmission path. The α, β, and γ are the intrinsic PLE for signals propagating through air, foliage, and ground. We collected RSSI data at four locations in an orchard across four weeks. Data from the first week were used to fit the values of α, β, and γ. Log was then evaluated over the next three weeks in two cases: 1) without updating α, β, and γ; 2) updating α, β, and γ using data from only one location.

Figure 6 shows the estimation error for both cases. The error in case 2 significantly reduced from 13.9 to 5.7 dB across all locations, compared to case 1. This is due to orchards' media homogeneity, where all locations share the same α, β, and γ values. Thus, updating Log with data from just one location effectively improves accuracy across others. *Hence, in orchards, updating models with data from a subset of locations enhances accuracy across others.*

3.4.2 Spatial Homogeneity. An orchard is divided into multiple areas, each covered by a gateway and maintaining a similar tree layout. Each gateway holds its own database for its coverage area. The similarity of CSI databases across areas, owing to tree layout uniformity, is spatial homogeneity.

To confirm this assertion, we analyze the spatial spectra of identical location IDs within two areas, using the PSNR (Peak Signal-to-Noise Ratio) to quantify their similarity. Higher PSNR values indicate greater similarity [43]. As illustrated in Figure 7, PSNR for the same location IDs across two areas reaches as high as 25.1 dB, suggesting high spectral similarities. Moreover, Figure 8 presents PSNR of spatial spectra for corresponding location IDs across six areas, revealing that 86.7% of the spectra have a PSNR above 20 dB. Similar spatial spectra imply similar CSI, as both reflect signals undergoing similar environmental impact. *This confirms that various areas within an orchard exhibit similar CSI fingerprints.*

4 WORKFLOW OF *ORCHLOC*

Figure 1 presents the workflow of *OrchLoc*, designed for efficient database building and updating in orchards, consisting of three stages: pre-training, fine-tuning, and inference.

In the pre-training stage, we select one area as the reference area, collecting CSI fingerprints from all locations within the reference area to build an initial database. This database pre-trains the CGM, which integrates a location-aware diffusion model with CSI and

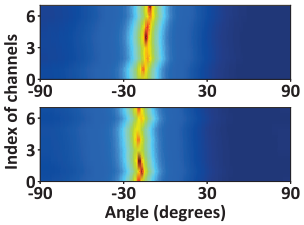


Figure 7: Spatial spectra of same ID in two areas.

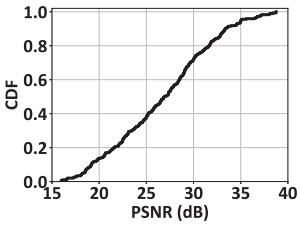


Figure 8: Similarity of spatial spectra across six areas.

FFZ-based location representers, enabling the generation of CSI fingerprints for different locations.

The fine-tuning stage is essential for either building or updating the database for an area. **Building:** Fingerprints from locations with static sensor nodes are used to fine-tune the pre-trained CGM, adapting it to local variations of the area. The refined CGM generates fingerprints for locations without sensors, which are then combined with data from sensor-equipped locations to form a complete database. This database are used to train the area-specific complex-valued location classifier. **Updating:** The distinction lies in fine-tuning the existing area-specific CGM instead of the initial pre-trained CGM.

During the inference stage, the trained classifier determines the robot's location based on its current CSI measurement.

The pre-training and fine-tuning stages, termed as turbo-training, leverages two observations made in orchards. First, using fingerprints from all locations in a reference area exploits *spatial homogeneity*, as this area's database characteristics mirror those of other areas. Second, the success of the fine-tuning across all locations is due to *Media Homogeneity*. Although CGM model is more complex than the Log model, its parameters essentially are akin to Log's α, β, and γ. These parameters represent the media's shadowing effect on the signal's RSSI or CSI in orchards. Thus, fine-tuned parameters of either Log or CGM are applicable across all locations.

5 CSI GENERATIVE MODEL

Figure 1 also outlines three components of the *CSI generative model (CGM)*. The location-aware diffusion model (LoDM), receiving input from CSI and location representers, learns the relationship between CSI fingerprints and location IDs. The CSI representer employs an complex-valued autoencoder: the encoder Φ transforms low-dimensional CSI into a high-order signal vector $\mathbf{z}_0$, while the decoder Ψ reconstructs the raw CSI $\tilde{\mathbf{x}}_0$ from the signal vector $\tilde{\mathbf{z}}_0$ generated by the LoDM. For the location representer, the location ID is converted to an FFZ vector $\mathbf{y}$ via FFZ modeling, which is then processed by an FFZ encoder Γ into a high-order link vector ζ.

Pairs of CSI fingerprints and their corresponding location IDs are fed into CGM for training. To generate CSI fingerprints, the location ID and Gaussian noise, mirroring the shape of a CSI fingerprint, are input into the well-trained CGM.

5.1 CSI Representer

Raw CSI inevitably contains measurement noise, which stems from factors like fluctuating atmospheric conditions. Directly using this noisy data can lead the LoDM to incorrectly interpret these transient disturbances as genuine features. Moreover, relying on a low-dimensional CSI vector might result in overfitting [44–46], where

the model becomes excessively tailored to specific patterns. This specialization can curb model's performance on new CSI data. To this end, we develop a complex-valued autoencoder to derive high-dimensional CSI representations.

The encoder Φ is comprised of three complex-valued FC blocks. Following each block, an *ReLU* function is applied to introduce non-linearity, thereby enabling the encoder to effectively capture higher-order representations within the CSI data. The dimension of each layer successively doubles that of the preceding one. Parallel to the encoder structure, the decoder Ψ is made up of three complex-valued FC blocks. It reconstructs the raw CSI data from the encoded representation. Starting with the high-dimensional size, each FC block in the decoder reduces the dimension by half, culminating in the reconstruction of the raw CSI vector $\tilde{\mathbf{x}}_0$.

After obtaining the reconstructed CSI vector $\tilde{\mathbf{x}}_0$ from the decoder Ψ, we compute the L_2 loss, given by $\mathcal{L}_{L_2}(\mathbf{x}_0, \tilde{\mathbf{x}}_0) = \|\mathbf{x}_0 - \tilde{\mathbf{x}}_0\|_2^2$, to train the complex-valued autoencoder.

5.2 Location Representer

The location representer incorporates FFZ modeling and an FFZ encoder to learn the location representation.

5.2.1 FFZ Modeling. The CSI vector is labeled with a location ID ranging from 1 to M, typically represented as a one-hot vector of length M, with a single bit set to 1 for the specific ID. However, one-hot encoding presents limitations, such as reduced training efficiency and lack of physical location context for guiding the LoDM in learning CSI-location relationships. To this end, FFZ modeling is devised. It transforms a location ID into a 7-element FFZ vector $\mathbf{y}$ that encapsulates physical factors influencing signal transmission.

The FFZ represents a 3D ellipsoid region concentrating most of the signal's energy, with focal points aligned to the 3D coordinates of the node and gateway. In orchards, the FFZ contains three media (air, foliage, and ground), each reporting distinct shadowing effects on the signal. Thus, we calculate the proportions of three media to profile the signal path.

A 3D orchard representation is created using a Cartesian coordinate system, with x and y axes along and across orchard rows, and z axis pointing upwards, originating at the gateway's position on the ground. Uniform tree spacing in orchards determines tree positions on x and y axes. Trees are modeled as cylinders (trunks) and ellipsoids (crowns), only requiring measurements of height and canopy width. Consistent growth patterns of trees across the same orchard allow for modeling based on a single representative tree, thus enabling a 3D representation of the orchard.

Utilizing the 3D coordinates of the FFZ's focal points and its mathematical formulation [42], the 3D FFZ is represented within the established coordinate system. Numerous sampling points within the 3D FFZ are then evaluated for interactions with air, trees, or ground. The proportion of each medium is calculated by comparing the number of sampling points interacting with each medium to the total within the 3D FFZ.

Furthermore, acknowledging the impact of both distance and direction between node and gateway on the received signal, the normalized distance and 3D direction are incorporated. This yields a 7-dimensional FFZ vector $\mathbf{y}$.

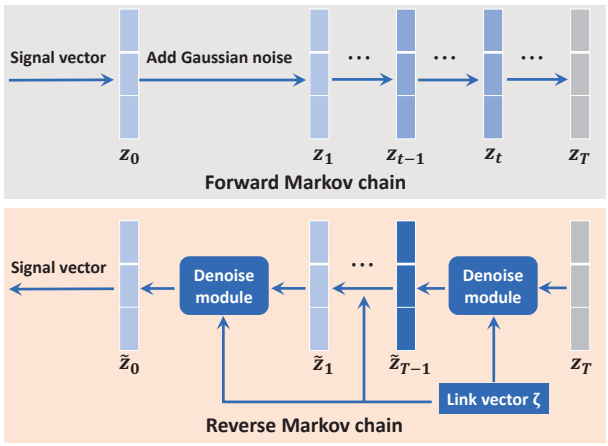


Figure 9: The location-aware diffusion model (LoDM).

5.2.2 FFZ Encoder. The FFZ vector $\mathbf{y}$ captures factors affecting the CSI vector, providing the LoDM with physical location information to understand the CSI-location relationship. However, due to uniform orchard layouts, identical FFZ vectors are generated for the same location ID across different areas, limiting new area adaptation for the LoDM. To overcome this, the FFZ encoder Γ transforms the FFZ vector $\mathbf{y}$ into the link vector ζ. Comprising multiple real-valued FC layers with *ReLU* activation, it is co-trained with the LoDM. Therefore, refining the FFZ Encoder while fine-tuning the LoDM to new areas allows it to adjust to subtle local environmental variations.

5.3 Location-Aware Diffusion Model

We develop the LoDM for generating CSI across various locations, inspired by the denoising diffusion probabilistic model (DDPM) [28],

5.3.1 Overview. Figure 9 depicts the architecture of LoDM, which consists of two Markov chains: forward and reverse. In the forward chain, unlike DDPM that introduces noise to raw data, we use signal vectors $\mathbf{z}_0$ (high-order representations of raw CSI vector $\mathbf{x}_0$) to produce a sequence of noisy signal vectors $\mathbf{z}_t$. This is achieved by incrementally adding Gaussian noise to $\mathbf{z}_0$ at each time step t, which ranges from 1 to T.

In the reverse chain, the critical task is to accurately subtract noise from the noisy signal vector via the denoise module. Considering location effects on the signal, the module inputs both the noisy signal vector $\mathbf{z}_t$ and the link vector ζ to estimate noise in $\tilde{\mathbf{z}}_t$ at each time step t. Attention layers are incorporated to fuse the signal and link vectors, enhancing the noise estimation process. Subtracting this estimated noise from noisy vector $\tilde{\mathbf{z}}_t$ produces $\tilde{\mathbf{z}}_{t-1}$, the input for the next denoise step $t-1$. This iterative process progresses from T to 1, ultimately recovering the initial signal vector $\tilde{\mathbf{z}}_0$.

The generation of a new single vector requires two inputs: a link vector ζ and a random noisy vector $\mathbf{z}_T$, sampled from a Gaussian distribution. This is followed by employing ancestral sampling through the reverse Markov chain.

5.3.2 Denoise Module. Figure 10 shows the denoise module, featuring a deep neural network-based noise predictor. It integrates a complex-valued U-Net, an embedding layer, and attention layers to estimate noise in the noisy signal vector $\tilde{\mathbf{z}}_t$, given time step t and link vector ζ. The module subtracts the estimated noise from $\tilde{\mathbf{z}}_t$,

generating $\tilde{\mathbf{z}}_{t-1}$ for the next step $t-1$. Thus, the noise predictor is central to the module's function.

Complex-Valued U-Net Backbone: The U-Net backbone [29], illustrated in Figure 10, features a dual-path design comprising contracting and expansive segments. This architecture excels in feature extraction and reconstruction [28], making U-Net well-suited for generating CSI fingerprints that capture signal variations induced by environmental conditions in orchards. The contracting path (right trapezoid) compresses the signal vector for key feature extraction and global trend identification. Conversely, the expansive path (left trapezoid) reconstructs this data, focusing on subtle details and maintaining spatial correlations. This dual architecture captures the complex dimensions of the signal vector.

In detail, the contracting path features downsampling layers with 1D convolutions (increasing channel count), *ReLU* activations, residual layers (dotted line), and complex-valued FC block reducing dimensions. The expansive path includes upsampling layers with 1D convolutions (decreasing channel count), *ReLU* activations, residual layers, and complex-valued FC block expanding dimensions.

Embedding Layer: The time step t is transformed from an integer into a vector using Sinusoidal Position Embeddings [30]. This embedding vector is subsequently added to both the input and intermediate layers of the predictor.

Attention Layer: Our noise predictor integrates the link vector ζ via an cross-attention layer [30, 31, 47]. This integration enables the model to focus on location-relevant features, enhancing its capacity to produce signal vectors for each location. The link vector ζ is effectively utilized in both downsampling and upsampling layers of U-Net via attention layers:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V \tag{2}$$

where $Q = W_Q \cdot \pi_i(\mathbf{z}_t)$, $K = W_K \cdot \zeta$, $V = W_V \cdot \zeta$ are computed using learnable projection matrices W_Q, W_K, W_V, and π_i is intermediate layer of U-Net. The scaling factor $\sqrt{d}$ maintains stability during the training process, where d is the feature size.

5.4 Generalizability to Other Environments

While our system theoretically holds the potential for application across diverse environments, such as wild forests, its generalizability depends on having an accurate 3D model of the environment for location representation and the presence of media and spatial homogeneity. These factors are crucial for constructing and updating the fingerprint database. The structured layout of orchards, featuring uniformly positioned trees along with media and spatial homogeneity, significantly facilitates efficient fingerprinting. In contrast, forests present a more complex scenario with trees varying in placement, species, age, and shape, posing substantial challenges to the system [48]. Such variability complicates the fingerprinting process, making it more labor-intensive. Future work could explore methods to adapt *OrchLoc* to other environments in a lightweight manner.

6 IMPLEMENTATION

Hardware: LoRa nodes are hand-crafted with SX1276 Radio [49] on the Arduino Uno boards [50]. The bladeRF 2.0 Software Defined Radio (SDR) [51] is used to receive LoRa signals. The SDR

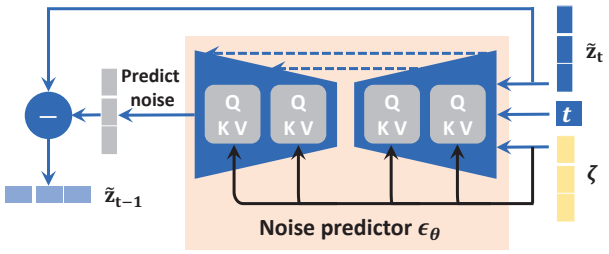


Figure 10: The architecture of the denoise module.

is equipped with two antennas, a common setting for LoRa gateways [52–54], producing two synchronized I and Q streams. The distance between two antennas is 14.0 cm, less than half the wavelength of LoRa signal. The SDR is connected to a Raspberry Pi 4. We execute Python scripts on the Raspberry Pi to control the SDR, enabling it to capture signal samples at a sampling rate of 2 MHz. While acknowledging that dual-antenna configurations are not standard in commercial LoRa gateways, our choice to use this setup with SDR is aimed at demonstrating its potential in addressing orchard localization challenges. We recognize that commercial gateways may not adopt this configuration. Our decision to utilize a gateway with two antennas explore and demonstrate the potential in addressing in-orchard localization challenges. Collected samples are processed on a local computer equipped with a CPU that has an Intel(R) Core(TM) i9-11900KF @ 3.50 GHz. A NVIDIA GEFORCE RTX 3080 Ti card is used to accelerate the training process. GPUs are not required for inference, making *OrchLoc* compatible with edge devices employed in agricultural applications.

Training of the Location Classifier: The input layer requires data in the format $\mathbf{X} = \{x_r, x_i\} \in \mathbb{R}^{l \times 2}$, with the first and second columns representing the real and imaginary parts of inputs, respectively. Thus, we flatten the 2×8 CSI vector into a 16-element column vector ($l = 16$), and then compute the real and imaginary values for each element. The output layer's size is equal to the number of locations M. We use cross-entropy loss [55] to calculate the loss between the actual and predicted location IDs.

Training of the CSI Generative Model: The CSI representer shares the same input as the location classifier. Its encoder generates the signal vector $\mathbf{z}_0$ with dimensions 2×128. The location representer, utilizing FFZ modeling, creates a 7-element FFZ vector $\mathbf{y}$. Then, the FFZ encoder outputs a link vector ζ of length 32. To reduce the risk of overfitting, a *dropout* layer with a probability of 0.1 is integrated into both the CSI representer and the FFZ encoder. The CSI representer is trained using the Adam optimizer, with a batch size of 256 and a learning rate of 0.001.

For LoDM model, Gaussian noise with a mean of zero and a variance β_t is introduced in the forward process. The variance β_t increases linearly from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$, where T is the total number of time steps in the diffusion process. We set the total number of time steps $T = 1000$. Its noise predictor is trained to minimize the discrepancy between the predicted and actual noise. Training continues until a specific convergence criterion is satisfied. This can be either reaching the maximum number of epochs or achieving a loss lower than a specified threshold. In our implementation, the maximum epoch is set to 300 and the minimum loss is 0.01. The chosen hyper-parameters align with those suggested by [28].

To enhance the noise predictor's learning, which depends on signal and link vector pairs, we employ the L_2 loss:

$$\mathcal{L}_{LoDM} = \mathbb{E}_{\Phi(\mathbf{x}_0),\mathbf{y},t,\epsilon\sim\mathcal{N}(0,1)}\left[\|\epsilon - \epsilon_\theta\left(\mathbf{z}_t, t, \Gamma\left(\mathbf{y}\right)\right)\|_2^2\right] \quad (3)$$

where $\mathbf{z}_t$ is the noisy version of $\mathbf{z}_0 = \Phi(\mathbf{x}_0)$ at time step t, with $\Gamma\left(\mathbf{y}\right)$ producing link vector ζ. Time step t is sampled uniformly from $\{1, \cdots, T\}$, ϵ represents the true added noise in $\mathbf{z}_t$, and $\epsilon_\theta\left(\mathbf{z}_t, t, \Gamma\left(\mathbf{y}\right)\right)$ is the predicted noise. Noise predictor ϵ_θ and FFZ encoder Γ are co-optimized using this loss.

7 EVALUATION

7.1 Experimental Setting

Pistachio orchards, an economic crop yielding approximately 1.1 billion pounds in 2021, were selected for our evaluation. Figure 11(a) showcases a pistachio orchard testbed divided into several areas, with each area containing $M = 64$ trees. Among these areas, area A served as the reference area. In this orchard, the arrangement features trees spaced 4.9 m apart in columns and 6.6 m between rows, with the average tree dimensions being 6.1 m in height and 2.3 m in width.

Within each area, LoRa nodes were positioned adjacent to 64 trees, with each node situated 2.0 m to the right of the trees. At each node location, LoRa nodes transmitted an 8-byte packet to the gateway. As shown in Figure 11(b), the LoRa nodes and gateway were installed at heights of 0.45 m and 10.0 m, respectively. The system was configured with a transmission power of 14 dBm, a spreading factor of 10, a bandwidth of 125 kHz, and a coding rate of 4/5. The successive eight packets employed eight different channels, cycling through sending the eight packets with different channels over 20 times. Consequently, a total of 160 packets were sent from each node location, yielding 20 CSI vectors per location. We also employ the data augmentation technique proposed in Section 3.1.1 to produce 80 additional CSI vectors at each location by adding Gaussian noise to the initial 20 CSI vectors. Thus, a total of 100 samples are collected at each location.

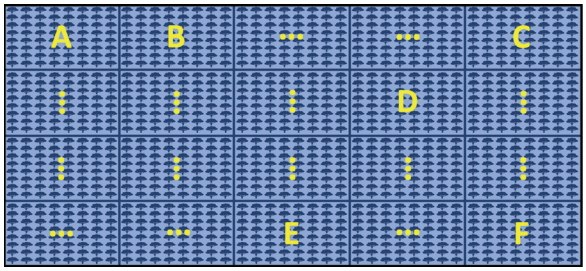
7.1.1 Datasets. In all areas, LoRa nodes are located 2.0 m right from trees. At each location, the node transmitted a packet to the gateway, cycling through eight channels for eight successive packets, repeated 20 times. Thus, 160 packets were sent from each location. Area A in Figure 11 is selected as the reference area.

Temporal Dimension Dataset: CSI fingerprints were collected in area A seven rounds ($r1$ to $r7$) over four weeks.

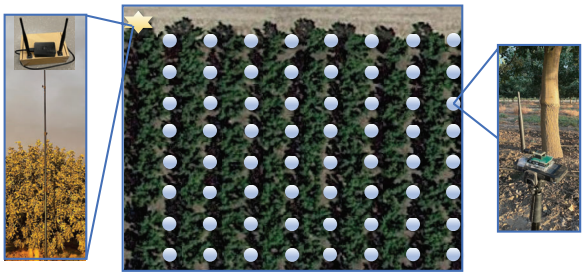
Spatial Dimension Dataset: We collected CSI data in six areas (A to F in Figure 11(a)) of one pistachio orchard and four areas of another pistachio orchard.

7.1.2 Benchmarks. We use the following benchmarks:

- *GPS:* We use Google Maps in satellite mode on an IC 5941 GPS radio[56] to identify current location IDs, then compare them with the actual physical locations.
- *RSSI:* RSSI data from eight channels are employed to identify locations via FC layers.
- *AMP:* Only the amplitude of CSI data is used for location identification via FC layers.
- *PHA:* Similarly, we only use the phase of CSI data to identify locations via FC layers.



(a) An orchard is divided into multiple areas.



(b) The setup in one area (one gateway and M locations).

Figure 11: The illustration of the orchard testbed setup.

- *KNN:* The CSI vector is flattened into a one-dimensional vector, with KNN applied as the classifier.

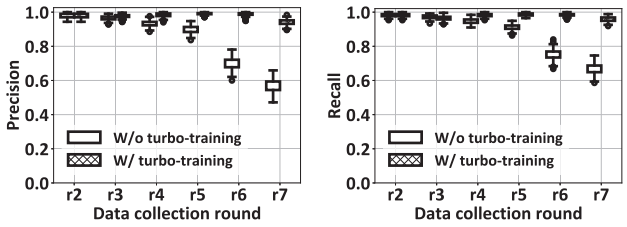
7.1.3 Performance Criteria. The location classifier assigns a CSI vector to a specific location, essentially a classification task. We evaluate *OrchLoc* based on average *precision* and *recall*, then translate classification results into localization error.

Precision and *recall* are calculated for each location: *precision* as the ratio of correctly identified instances to all predictions for that location, and *recall* as the ratio of correct identifications to the total instances at that location. These two metrics serve as indicators of the classifier's accuracy and reliability in correctly identifying tree locations. The emphasis on *precision* and *recall* can vary depending on specific applications. For instance, in the case of disease detection among trees, a high *recall* is paramount. This ensures that all potentially diseased trees are flagged for further inspection, prioritizing the identification of all cases over the risk of false positives. Conversely, other applications might demand higher *precision* to minimize the risk of false identifications.

Localization error is the Euclidean distance between the actual and predicted location IDs, taking into account the row and column spacing. Here, each tree is assigned a unique location ID, with a location point situated 2.0 m to the right of the tree's trunk.

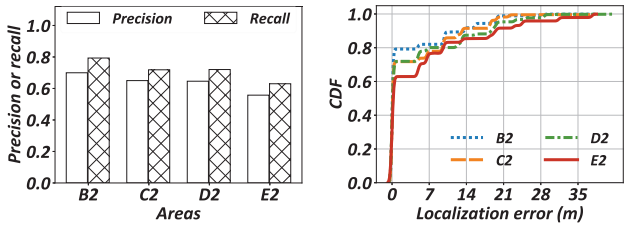
7.2 Temporal Dimension Performance

The efficacy of our system may be affected by various environmental dynamics, which can be classified into three categories [4]: 1) short-term weather changes, *e.g.*, temperature fluctuations; 2) changes in foliage density; and 3) long-term changes in foliage shape. Our temporal dataset encompasses the first two dynamics. Specifically, weather variation across the seven rounds is detailed in Table 1, and the rapid growth of pistachio trees in July suggests changes in foliage density [57]. For each data collection round, CSI data



(a) Precision (b) Recall

Figure 12: The performance in area A from rounds $r2$ to $r7$ without and with the turbo-training scheme.



(a) Classification error. (b) Localization error.

Figure 13: The localization accuracy in the four areas of the second pistachio orchard.

are collected at $M = 64$ locations, with the data from each location divided into training and testing parts in a $7:3$ ratio. Thus, for each round, the training or testing parts for all locations collectively forms the round's training or testing dataset.

In response to these environmental variability, we devised a turbo-training scheme that autonomously refreshes the fingerprints. First, a CGM is pre-trained with the round $r1$ training set, necessitating manual collection of CSI data at each location. Following this initial manual measurement, the CGM is fine-tuned using data from sensor nodes [58] already deployed for other agricultural tasks (*e.g.*, smart irrigation [59] and pest monitoring). To simulate data from these locations with sensor nodes, we randomly select training parts from 30% of the locations at each round for fine-tuning the pre-trained CGM. The refined CGM is capable of generating updated fingerprints for every location at the subsequent round, thus maintaining the classifier's efficacy. The refreshed classifier is then applied to identify location IDs from the CSI data in each round's testing set.

As outlined in Table 1, temperature fluctuations are confined to a 3.0°C range over a period of four hours. If a sensor node transmits its sensing data every 15 minutes, this enables a gateway to collect 16 CSI samples from a single sensor node within four hours. Before encountering significant temperature changes, our turbo-training scheme capitalizes on these samples to fine-tune the CGM.

Figure 12 shows that, with turbo-training, our classifier achieves an average *precision* of 96.3% and *recall* of 97.6%, alongside a localization error of 0.4 m. This demonstrates resilience to temperature variations up to 10.5°C, significantly surpassing the threshold of 3.1°C identified in Section 3.3. Thus, turbo-training effectively manages short-term weather changes and foliage density variations, eliminating the need for manual CSI fingerprint collection.

Moving forward, if temperature deviations exceed 10.5°C, our system's performance may decline. Future endeavors will aim at thoroughly evaluating and enhancing our system's robustness through extensive experimental studies. Moreover, while rapid pistachio growth was observed, it did not cover a wide range of foliage changes. Significant alterations in tree foliage, such as extensive leaf loss impacting CSI features, could potentially affect our system's performance adversely. Our system is particularly beneficial in environments with dense foliage, where thick foliage can significantly obstruct GPS signals. Conversely, in seasons with sparse foliage, such as late autumn or winter, simpler GPS/INS systems may provide sufficient navigation accuracy due to unobstructed GPS signal paths [9], thereby reducing the necessity for our system.

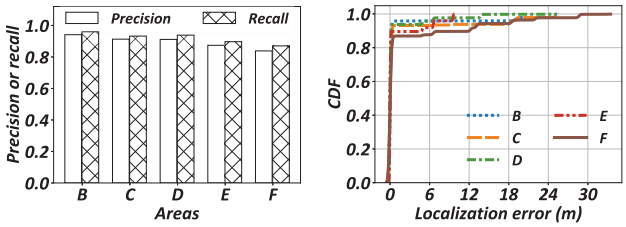
Long-Term Foliage Shape Changes: The foliage shape changing, characterized by trunk and branch growth over years, necessitates an evaluation of its impact on localization accuracy. To this end, data were collected in a second pistachio orchard exhibiting slightly distinct characteristics from the first: row and column spacings of 6.6 m and 4.8 m, respectively, and average tree dimensions of 5.8 m in height and 2.1 m in width. This variance in orchard layout and tree size provides an ideal context for assessing the adaptability of our system to long-term environmental changes. Utilizing the pre-trained CGM from the reference area, we can generate fingerprints for new areas, whether they are areas covered by another gateway within the first orchard or entirely different orchards. This adaptability is enhanced by fine-tuning the pre-trained CGM with packets received from existing sensor nodes in the new area, tailoring the model to the local environmental conditions.

CSI data were collected from four areas ($B2$, $C2$, $D2$, and $E2$) in the second orchard. For each area, training data from 30% of the locations were randomly selected to refine their classifiers through the turbo-training scheme, employing the CGM pre-trained with data from area A. The performance of these updated classifiers was then evaluated in their corresponding areas to gauge their adaptability to variations in foliage structure. It is imperative to note that the collected data at different areas is only for simulating data from locations having sensor nodes and for system evaluation purposes.

As depicted in Figure 13, our system demonstrated commendable performance in the second orchard, achieving *recall* rates of 79.3%, 71.9%, 72.1%, and 63.1%, and mean localization errors of 2.8, 3.5, 4.1, and 5.3 m across the four areas. Although a performance decrement was observed, likely attributable to the differences between the orchards, the results remain promising. With an average *precision* and *recall* of 63.8% and 71.6%, respectively—without necessitating manual data collection—the achieved accuracy surpasses the inter-tree distance of 6.6 m and the typical GPS error of 7.9 m (as detailed in Section 7.4). These findings validate our system's potential to significantly reduce labor costs for training data collection in new orchards, underscoring its adaptability to environmental changes.

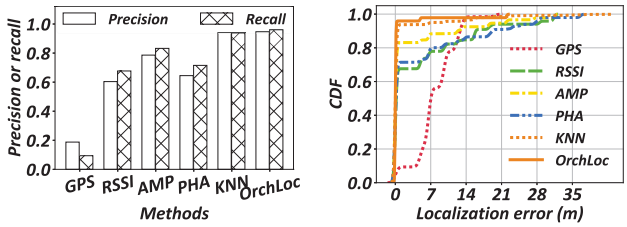
7.3 Spatial Dimension Performance

The turbo-training scheme was employed across five new areas (B, C, D, E, and F) within the first orchard, selecting a random 30% of locations in each for this turbo-training. This facilitated the development of location classifiers tailored to each specific area, which were then utilized for testing in these areas.



(a) Classification error. (b) Localization error.

Figure 14: The localization accuracy in the five areas of the first pistachio orchard.



(a) Classification error. (b) Localization error.

Figure 15: The localization accuracy in area A with six different location classifier.

As depicted in Figure 14, our system demonstrated substantial effectiveness in these new areas. Notably, we attained *precision* of 95.9%, 93.3%, 93.9%, 89.7%, and 87.1%, along with mean localization errors of 0.5, 0.7, 0.8, 1.3, and 2.1 m for areas B, C, D, E, and F respectively. A discernible trend is the gradual performance decline correlating with increased distances from area A. This pattern is likely attributed to shifts in the data distribution resulting from localized environmental variations across different areas when compared to the reference area A. Additionally, the discrepancies in accuracy relative to those presented in Figure 12 could stem from both spatial and temporal diversity—given that the data from these new areas were collected at different times relative to area A. Incorporating additional location data for turbo-training could mitigate this.

7.4 Benchmark Study

We assess five baselines on the dataset from area A at round $r1$. The RSSI-based, amplitude-based, and phase-based classifiers utilize three FC layers. The amplitude and phase data are derived from CSI data, while RSSI is calculated using collected I Q samples. For KNN-based classifier, the location ID for a CSI vector in the test dataset is identified by finding the k nearest CSI vectors in the training dataset via Euclidean distance, and assigning the most common ID among these neighbors. After a grid search, we selected $k = 5$.

Currently, smartphones utilize signals from multiple Global Navigation Satellite Systems (GNSS), including GPS, GLONASS, and Galileo, combining these to provide fused positioning results. Despite this integration of several signals, Figure 15 shows that GPS has the lowest *precision* and *recall*, at 18.8% and 9.4% respectively, with a mean localization error of 7.9 m. It is due to dense tree canopies obstructing GPS signals. Instead of correctly pinpointing the location ID, GPS often identifies neighboring rows or columns.

OrchLoc outperforms *RSSI*, *AMP*, and *PHA*, enhancing *precision* by 56.6%, 20.3%, and 46.7%, and *recall* by 41.9%, 15.4%, and 34.3%, respectively. The mean localization errors using RSSI, amplitude, phase, or CSI are 4.6, 2.7, 4.7, and 0.5 m, highlighting CSI's superiority in location identification. While amplitude alone is limited, the incorporation of phase information enriches location matching. Our complex-valued classifier, unlike simple FC layers, effectively utilizes the full potential of the CSI data, boosting accuracy.

Although a marginal improvement of *OrchLoc*, compared to KNN, is observed with a 2.0% increase in *precision* and a 3.9% increase in *recall*, our system outperforms KNN in several aspects. First, as illustrated in Figure 15(b), our classifier significantly reduces the mean localization error from KNN's 0.99 m to 0.54 m, translating to a 45.0% improvement. Upon a detailed examination of all instances of misclassification, we observed that the mean localization errors for KNN and our system were 7.8 m and 14.6 m, respectively. Despite these comparable misclassification rates, our model consistently generates predictions that are closer to the true positions than those produced by KNN. This advantage is attributed to our classifier's complex-valued FC block for analyzing the interaction between phase and amplitude within CSI data. In contrast, KNN's method of merging amplitude and phase into a single column vector may obscure spatial or structural characteristics of the CSI vector, thereby interfering with the precise interpretation of the data.

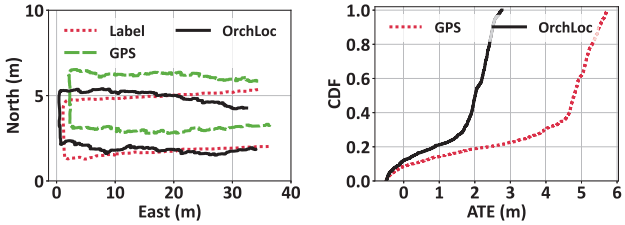
Second, for robotic navigation, the precision with which our classifier provides localization information is of utmost importance for calibrating INS drift. The significant reduction in localization error facilitated by our system necessitates less time for the EKF to correct drift, thus enhancing the efficacy of navigational adjustments. The lower precision and recall of KNN, coupled with its larger localization errors in misclassification cases, increase the risk of substantial navigational inaccuracies.

Furthermore, our classifier exhibits a remarkable improvement in computational efficiency for inference, requiring significantly fewer floating-point operations per second (FLOPs), specifically 34,816, compared to KNN's 6,850,160 FLOPs. This drastic difference in computational demand can be attributed to the inherent characteristics of the algorithms; KNN, being a lazy learning algorithm, lacks a training phase and necessitates the comparison of all instances in the training dataset during inference to identify the most similar instances. In contrast, our classifier efficiently performs a single forward computation for inference, thereby streamlining the computational process and reducing the operational burden.

Finally, it is also critical to emphasize that both KNN and our classifier benefit from our innovative turbo-training scheme, incorporating the CGM for refreshing fingerprints. This approach enables both algorithms to utilize updated fingerprints, thereby ensuring accurate and reliable classification results.

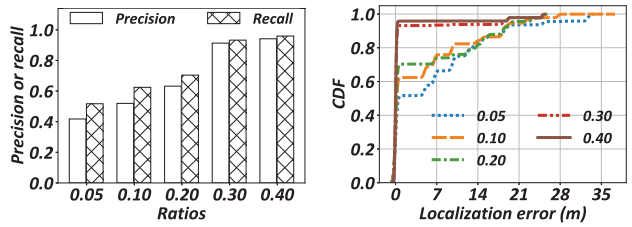
7.5 Robot Navigation

To assess the performance of *OrchLoc* in the context of robot navigation, we employ the public dataset [9], as outlined in Section 2. This dataset enables us to simulate a robot's path through an orchard environment. The dataset is formatted as a time series, comprising INS data and corresponding ground truth coordinates for each timestamp. The simulation process initiates by aligning the robot's starting position with a tree, ensuring the robot's initial direction of movement



(a) Trajectory reconstruction (b) Absolute trajectory error (ATE)

Figure 16: Comparison of robot navigation performance in an orchard: GPS *vs. OrchLoc* computed with INS data.



(a) Classification error. (b) Localization error.

Figure 17: Impact of ratios of number of locations.

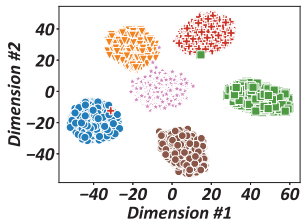


Figure 18: Impact of the positions of the LoRa nodes.

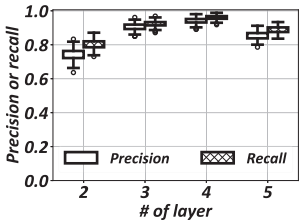


Figure 19: Impact of the number of sampling layer.

is parallel to the rows of trees. To incorporate CSI data into each moment of the simulation, we identify the tree closest to the robot's current coordinate and extract all available CSI measurements associated with that tree. We randomly select one CSI measurement to represent the timestamp, mirroring the process of integrating GPS data. These CSI data points are subsequently fed into *OrchLoc* to derive inferred location coordinates. To mimic the real-world scenario where a robot might not align perfectly with a tree's location, random noise is introduced to the inferred coordinates.

The integration of INS data with GPS or *OrchLoc* is achieved through the state-of-the-art Neural-KF algorithm [9]. Figure 16(a) illustrates two reconstructed trajectories, highlighting *OrchLoc*'s enhanced accuracy in comparison to GPS-based navigation. As depicted in Figure 16(b), *OrchLoc* consistently achieves lower navigation errors, with an average error of 1.2 m compared to GPS's 3.1 m. These findings underscore the potential of *OrchLoc* to significantly improve navigation accuracy in environments where GPS signals are obstructed or unreliable.

It is important to recognize that the higher GPS accuracy observed in Figure 16 compared to Figure 2 stems from the utilization of distinct sources of GPS data. In this section, GPS data corresponds to location IDs, specifically tree locations, which were identified using Google Maps in satellite mode while navigating within orchards. These location IDs are then translated into their respective physical locations, positioned 2.0 m to the right of the tree's trunk, incorporating tree location priors into the GPS data. Conversely, in Figure 2, the GPS data originates from the public dataset [9], to which we applied an average error of 7.9 m to realistically simulate GPS performance in orchard environments.

7.6 Parameter Study

In this section, unless specified otherwise, turbo-training scheme involves pre-training CGM in area *A* and fine-tuning it with 30% of the location data from area *B*. The accuracy of the trained classifier is tested on the testing set of area *B*.

7.6.1 Ratios of Fine-Tuning Data. We explored varying the fine-tuning location ratios from 5% to 40%. As Figure 17 illustrates, performance improves with an increased number of locations for fine-tuning the CGM. *OrchLoc* achieves 94.1% *precision*, 95.9% *recall*, and an average localization error of 0.8 m in area *B* when the ratio exceeds 30%. However, ratios below 20% yield lower results, such as only 51.9% *precision* at a 10% ratio.

OrchLoc's requirement to deploy LoRa sensor nodes at 30% of trees complements precision agriculture's evolving demands [1, 2, 60]. Future efforts will focus on preserving accuracy while reducing the number of necessary deployment locations through: 1) Incorporating additional data types like satellite imagery; 2) Utilizing active learning to intelligently select critical deployment sites; 3) Investigating hybrid models combining supervised and unsupervised techniques to potentially enhance accuracy with fewer sites.

7.6.2 Position of LoRa Node. Typically, the LoRa node is placed 2.0 m right to each tree during CSI data collection. To evaluate the impact of node position offset on CSI data distribution, the node was placed at five different locations around the predetermined spot for each tree, repeated for six trees. Figure 18 presents the CSI data visualized via t-SNE [41]. The data exhibit a clear clustering effect, with data from the same tree clustering together, while those from different trees are distinct. This indicates that, even though there is a position offset at each location, the data collected close to each tree still exhibit a similar distribution.

7.6.3 Number of Sampling Layers in Noise Predictor. The noise predictor in *OrchLoc* includes multiple sampling layers within its contracting and expansive paths. We varied the number of these layers from 2 to 5 in our experiments. Figure 19 shows that increasing the number of layers from 2 to 4 enhances *precision* from 74.2% to 94.1%. Yet, further increasing to 5 layers drops *precision* to 85.2%. While fewer layers may not adequately extract latent CSI-location relationships, too many layers can excessively reduce the vector length, potentially leading to information loss.

7.7 Overhead of *OrchLoc*

Sensory LoRa Node: LoRa nodes deployed in orchards for agricultural tasks transmit eight packets across eight channels. Segmenting sensing data for transmission over individual channels minimizes additional energy usage. While this increases collision risk, it can be mitigated through careful scheduling, utilizing the week-long high

accuracy of the location classifier, and employing efficient MAC protocols or collision resolution techniques [34, 61–63].

Robot: For a single localization inference, the robot transmits eight two-byte packets and receives the localization result from the gateway, a process that consumes 1.4 seconds at SF10 (the lowest data rate permitted in the US). Given the robot's velocity of 0.3 m/s [64], typical of agricultural robots, this translates to a movement of 0.42 m within the 1.4-second interval. As depicted in Figure 18, such a minor positional shift exerts a negligible impact on CSI data quality. For robots operating at higher speeds, the strategy of incorporating periodic 1.4-second pulses for localization purposes can effectively minimize potential accuracy degradation. Regarding power consumption, with a power of 0.4 W over a duration of 1.4 s, the energy consumption is 0.56 J per localization inference. Consequently, *OrchLoc* is a low-power, sustainable solution well-suited for agricultural deployments.

8 RELATED WORK

Robot Navigation: GPS/INS combines INS data with GPS data using a Kalman Filter-based algorithm for robot navigation [9, 14]. For instance, Neural-KF [9] utilizes a neural network [65, 66] to estimate the robot's velocity and location from raw inertial data, then refining these estimates with GPS data via a Kalman filter to improve navigation accuracy. These algorithms critically rely on the precision of GPS data to compensate for the INS sensor drift. However, in orchard environments, the accuracy of GPS data is notably compromised due to signal blockage by crop canopies, presenting a considerable challenge [9, 14]. While Real-Time Kinematic (RTK) GPS offers enhanced localization accuracy, their deployment necessitates significant infrastructure investment, including the installation of base stations throughout the orchard. Such requirements render RTK GPS economically impractical for a wide range of agricultural operations. In contrast, our system seeks to utilize the pre-existing LoRa network infrastructure within orchards to provide a robust and cost-effective positional reference for INS sensors.

Localization by LoRa Network: The limited bandwidth of LoRa, at only 125 kHz [67, 68], poses challenges to directly applying conventional localization algorithms, which are typically designed for Wi-Fi, Bluetooth, or cellular networks, due to their reliance on higher bandwidths [5, 69, 70]. For instance, cellular-based localization often requires densely deployed base stations, which is impractical in agricultural settings where cell tower density is notably low, thus diminishing accuracy. Moreover, the data transmission and service fees render them cost-prohibitive for agriculture.

Current research on localizing LoRa nodes predominantly employs techniques such as Time Difference of Arrival (TDoA) [21, 22], Angle of Arrival (AoA) [23], or path loss models [24]. For example, Seirios [23] achieves a median localization error of 4.4 m across a $6,000\ m^2$ area employing AoA-based localization with a dual-antenna gateway. However, such methods typically necessitate a minimum of three gateways and rely on the existence of a LoS signal path from the node to the gateway, conditions that are often unmet in orchard environments. In contrast, our system introduces a fingerprinting-based approach to localization in orchards utilizing a single gateway, circumventing the limitations associated with multi-gateway setups and direct signal path dependencies.

Existing LoRa localization works use RSSI as fingerprint [71, 72]. They match RSSI measurements from multiple gateways with a database to determine a node's location. However, such methods presuppose that a packet is received by multiple gateways, a scenario infeasible in orchards. In contrast, *OrchLoc* enables in-orchard localization using only one gateway for extracting CSI fingerprints.

CSI-based Fingerprinting in Wi-Fi: CSI-based fingerprinting is developed in Wi-Fi for indoor localization [25–27]. These methods utilize either amplitude or phase independently, overlooking the interconnected information between them. *OrchLoc* introduces a complex-valued classifier to effectively learn the latent information among amplitude and phase of CSI data. Future works may extend to non-linear chirps to enhance signal heterogeneity [73] and could be tested for localizing multiple transmitters simultaneously [74–76].

Crowdsourcing has been widely used to gather fingerprints for WiFi-based indoor localization [77, 78]. For instance, Zee [77] utilizes smartphones' inertial sensors to map fingerprints onto indoor maps during users' routine walks. However, crowdsourcing is impractical in orchards, as it requires numerous robots to navigate within the orchard and accurate tracking of the robots' trajectories, both of which are challenging to achieve.

Generative Model for Wireless Signal: Recent studies have employed generative models for wireless signal modeling [43, 79, 80]. For example, NeRF^2 [79] segments 3D spaces into many small voxels for signal strength estimation. However, it is impractical in large orchards due to the enormous number of voxels required for processing. WiNeRT [80] computes the received signal based on environmental meshes, which are unavailable in orchards. Large language models reveal strong capability in code generation and serial understanding [81–83] but fall short in interpreting and generating wireless signals [43]. This paper customizes diffusion models for the synthesis of CSI data across various locations.

9 CONCLUSION

We introduce *OrchLoc* for in-orchard fingerprinting-based localization system that achieves tree-level localization accuracy with a single gateway. We propose CSI-based fingerprint and a complex-valued location classifier for location estimation. A turbo-training scheme, powered by the CGM, is devised to efficiently build and update fingerprint database for each area, significantly reducing labor costs. Extensive experiments validate the efficacy of *OrchLoc*.

ACKNOWLEDGMENTS

This publication was prepared with the support of a financial assistance award approved by the Economic Development Administration, Farms Food Future. It was also supported in part by NSF Grant #2239458, a UC Merced Fall 2023 Climate Action Seed Competition grant, and a UC Merced Spring 2023 Climate Action Seed Competition grant. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

A ARTIFACT

The research artifacts accompanying this paper are available via https://zenodo.org/doi/10.5281/zenodo.11088623.

REFERENCES

[1] Robin Gebbers and Viacheslav I Adamchuk. Precision agriculture and food security. *Science*, 327(5967):828–831, 2010.

[2] Rodolfo Bongiovanni and Jess Lowenberg-DeBoer. Precision agriculture and sustainability. *Precision agriculture*, 5:359–387, 2004.

[3] Xiuhua Li, Won Suk Lee, Minzan Li, Reza Ehsani, Ashish Ratn Mishra, Chenghai Yang, and Robert L Mangan. Spectral difference analysis and airborne imaging classification for citrus greening infected trees. *Computers and Electronics in Agriculture*, 83:32–46, 2012.

[4] Kang Yang, Yuning Chen, Xuanren Chen, and Wan Du. Link Quality Modeling for LoRa Networks in Orchards. In *ACM/IEEE IPSN*, 2023.

[5] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. mD-Track: Leveraging multi-dimensionality for passive indoor Wi-Fi tracking. In *ACM MobiCom*, 2019.

[6] Yidong Ren, Li Liu, Chenning Li, Zhichao Cao, and Shigang Chen. Is lorawan really wide? fine-grained lora link-level measurement in an urban environment. In *IEEE ICNP*, 2022.

[7] Jingang Yi, Hongpeng Wang, Junjie Zhang, Dezhen Song, Suhada Jayasuriya, and Jingtai Liu. Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation. *IEEE transactions on robotics*, 25(5):1087–1097, 2009.

[8] Yaxuan Yan, Baohua Zhang, Jun Zhou, Yibo Zhang, and Xiao'ang Liu. Real-Time Localization and Mapping Utilizing Multi-Sensor Fusion and Visual–IMU–Wheel Odometry for Agricultural Robots in Unstructured, Dynamic and GPS-Denied Greenhouse Environments. *Agronomy*, 12(8):1740, 2022.

[9] Yayun Du, Swapnil Sayan Saha, Sandeep Singh Sandha, Arthur Lovekin, Jason Wu, S Siddharth, Mahesh Chowdhary, Mohammad Khalid Jawed, and Mani Srivastava. Neural-Kalman GNSS/INS Navigation for Precision Agriculture. In *IEEE ICRA*, 2023.

[10] Martin Brossard, Axel Barrau, and Silvere Bonnabel. RINS-W: Robust inertial navigation system on wheels. In *IEEE/RSJ IROS*, 2019.

[11] Yayun Du, Bhrugu Mallajosyula, Deming Sun, Jingyi Chen, Zihang Zhao, Mukhlesur Rahman, Mohiuddin Quadir, and Mohammad Khalid Jawed. A low-cost robot with autonomous recharge and navigation for weed control in fields with narrow row spacing. In *IEEE/RSJ IROS*, 2021.

[12] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, and Cyrill Stachniss. OverlapNet: Loop closing for LiDAR-based SLAM. *arXiv preprint arXiv:2105.11344*, 2021.

[13] Honghui Qi and John B Moore. Direct Kalman filtering approach for GPS/INS integration. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2):687–693, 2002.

[14] Nived Chebrolu, Philipp Lottes, Thomas Läbe, and Cyrill Stachniss. Robot localization based on aerial images for precision agriculture tasks in crop fields. In *IEEE ICRA*, 2019.

[15] Jansen C Liando, Amalinda Gamage, Agustinus W Tengourtius, and Mo Li. Known and unknown facts of LoRa: Experiences from a large-scale measurement study. *ACM Transactions on Sensor Networks*, 15(2):1–35, 2019.

[16] Yidong Ren, Wei Sun, Jialuo Du, Huaili Zeng, Younsuk Dong, Mi Zhang, Shigang Chen, Yunhao Liu, Tianxing Li, and Zhichao Cao. Demeter: Reliable Cross-soil LPWAN with Low-cost Signal Polarization Alignment. In *ACM MobiCom*, 2024.

[17] Xianzhong Ding and Wan Du. DRLIC: Deep Reinforcement Learning for Irrigation Control. In *ACM/IEEE IPSN*, 2022.

[18] DI Sacaleanu and IK Kiss. Monitoring walnut orchards with LoRa technology. In *IEEE SIITME*, 2018.

[19] Michele Preti, François Verheggen, and Sergio Angeli. Insect pest monitoring with camera-equipped traps: strengths and limitations. *Journal of pest science*, 94(2):203–217, 2021.

[20] Haozhe Zhang, Long He, Francesco Di Gioia, Daeun Choi, Antonio Elia, and Paul Heinemann. LoRaWAN based internet of things (IoT) system for precision irrigation in plasticulture fresh-market tomato. *Smart Agricultural Technology*, 2:100053, 2022.

[21] Dongfang Guo, Chaojie Gu, Linshan Jiang, Wenjie Luo, and Rui Tan. ILLOC: In-hall localization with standard lorawan uplink frames. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1):1–26, 2022.

[22] Semtech. Locating End Devices with Semtech's LoRa Cloud Geolocation Service. https://lora-developers.semtech.com/documentation/tech-papers-and-guides/locating-end-devices-with-lora-cloud/, 2021. [Online].

[23] Jun Liu, Jiayao Gao, Sanjay Jha, and Wen Hu. Seirios: leveraging multiple channels for LoRaWAN indoor and outdoor localization. In *ACM MobiCom*, 2021.

[24] Yuxiang Lin, Wei Dong, Yi Gao, and Tao Gu. Sateloc: A virtual fingerprinting approach to outdoor LoRa localization using satellite images. *ACM Transactions on Sensor Networks*, 17(4):1–28, 2021.

[25] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. CSI-based fingerprinting for indoor localization: A deep learning approach. *IEEE transactions on vehicular technology*, 66(1):763–776, 2016.

[26] Xuyu Wang, Lingjun Gao, and Shiwen Mao. CSI phase fingerprinting for indoor localization with a deep learning approach. *IEEE Internet of Things Journal*, 3(6):1113–1123, 2016.

[27] Xuyu Wang, Xiangyu Wang, and Shiwen Mao. Deep convolutional neural networks for indoor localization with CSI images. *IEEE Transactions on Network Science and Engineering*, 7(1):316–327, 2018.

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[31] Kang Yang, Xi Zhao, Jianhua Zou, and Wan Du. ATPP: A Mobile App Prediction System Based on Deep Marked Temporal Point Processes. *ACM Transactions on Sensor Networks*, 19(3):1–24, 2023.

[32] Natalia Piotrowski. Implementation and characterization of commercial off-the-shelf inertial measurement units for the attitude determination system of the move-iii cubesat. *Small Satellite Conference*, 2022.

[33] Joshua S Furtado, Hugh HT Liu, Gilbert Lai, Herve Lacheray, and Jason Desouza-Coelho. Comparative analysis of optitrack motion capture systems. In *Springer SMRC*, 2019.

[34] Xianjin Xia, Ningning Hou, Yuanqing Zheng, and Tao Gu. PCube: scaling LoRa concurrent transmissions with reception diversities. *ACM Transactions on Sensor Networks*, 18(4):1–25, 2023.

[35] Xianjin Xia, Qianwu Chen, Ningning Hou, Yuanqing Zheng, and Mo Li. XCopy: Boosting Weak Links for Reliable LoRa Communication. In *ACM MobiCom*, 2023.

[36] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. Jamming of LoRa PHY and countermeasure. *ACM Transactions on Sensor Networks*, 19(4):1–27, 2023.

[37] Yaxiong Xie, Zhenjiang Li, and Mo Li. Precise power delay profiling with commodity WiFi. In *ACM MobiCom*, 2015.

[38] Jun Liu, Weitao Xu, Sanjay Jha, and Wen Hu. Nephalai: towards LPWAN C-RAN with physical layer compression. In *ACM MobiCom*, 2020.

[39] Danijar Hafner, Dustin Tran, Timothy Lillicrap, Alex Irpan, and James Davidson. Noise contrastive priors for functional uncertainty. In *UAI*, 2020.

[40] Monson H Hayes. *Statistical digital signal processing and modeling*. John Wiley & Sons, 1996.

[41] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

[42] Theodore S Rappaport et al. *Wireless communications: principles and practice*, volume 2. prentice hall PTR New Jersey, 1996.

[43] Xingyu Chen and Xinyu Zhang. RF Genesis: Zero-Shot Generalization of mmWave Sensing through Simulation-Based Data Synthesis and Generative Diffusion Models. In *ACM SenSys*, 2023.

[44] Lorenzo Torresani and Kuang-chih Lee. Large margin component analysis. In *NeurIPS*, 2006.

[45] Zongxing Xie, Hanrui Wang, Song Han, Elinor Schoenfeld, and Fan Ye. DeepVS: A deep learning approach for RF-based vital signs sensing. In *ACM BCB*, 2022.

[46] Zongxing Xie, Ava Nederlander, Isac Park, and Fan Ye. Short: RF-Q: Unsupervised Signal Quality Assessment for Robust RF-based Respiration Monitoring. In *ACM/IEEE CHASE*, 2023.

[47] Kang Yang, Xi Zhao, Jianhua Zou, and Wan Du. ATPP: A Mobile App Prediction System Based on Deep Marked Temporal Point Processes. In *IEEE DCOSS*, 2021.

[48] Silvia Demetri, Gian Pietro Picco, and Lorenzo Bruzzone. LaPS: LiDAR-assisted placement of wireless sensor networks in forests. *ACM Transactions on Sensor Networks*, 15(2):1–40, 2019.

[49] Semtech. Semtech SX1276 Datasheet. https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276, 2020. [Online].

[50] Arduino. Arduino Uno Rev3. https://store-usa.arduino.cc/products/arduino-uno-rev3/?selectedStore=us, 2021. [Online].

[51] Nuand. bladeRF 2.0 micro xA4. https://www.nuand.com/product/bladerf-xa4, year = 2023, note=[Online].

[52] Raghav Subbaraman, Yeswanth Guntupalli, Shruti Jain, Rohit Kumar, Krishna Chintalapudi, and Dinesh Bharadia. BSMA: scalable LoRa networks using full duplex gateways. In *ACM MobiCom*, 2022.

[53] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. Don't miss weak packets: Boosting LoRa reception with antenna diversities. *ACM Transactions on Sensor Networks*, 19(2):1–25, 2023.

[54] Shiming Yu, Xianjin Xia, Ningning Hou, Yuanqing Zheng, and Tao Gu. Revolutionizing LoRa Gateway with XGate: Scalable Concurrent Transmission across Massive Logical Channels. In *ACM MobiCom*, 2024.

[55] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134:19–67, 2005.

[56] Dungmobile. 5941 IC GPS. http://dungmobile.vn/5941-ic-gps-iphone-xs-max-id3247.html, 2020. [Online].

[57] American Pistachio Growers. Growing and harvesting American pistachios. https://americanpistachios.org/growing-and-harvesting.

[58] Wan Du, Jansen Christian Liando, Huanle Zhang, and Mo Li. When pipelines meet fountain: Fast data dissemination in wireless sensor networks. In *ACM SenSys*, 2015.
[59] Xianzhong Ding and Wan Du. Optimizing Irrigation Efficiency using Deep Reinforcement Learning in the Field. *arXiv preprint arXiv:2304.01435*, 2023.
[60] Jian Ding and Ranveer Chandra. Towards low cost soil sensing using Wi-Fi. In *ACM MobiCom*, 2019.
[61] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. LMAC: Efficient carrier-sense multiple access for LoRa. In *ACM MobiCom*, 2020.
[62] Kang Yang and Wan Du. LLDPC: A Low-Density Parity-Check Coding Scheme for LoRa Networks. In *ACM SenSys*, 2022.
[63] Kang Yang, Miaomiao Liu, and Wan Du. RALoRa: Rateless-Enabled Link Adaptation for LoRa Networking. *IEEE/ACM Transactions on Networking*, 2024.
[64] IA Starostin, AV Eshchin, and SA Davydova. Global trends in the development of agricultural robotics. *IOP Conference Series: Earth and Environmental Science*, 1138(1):012042, 2023.
[65] Zhiyu An, Xianzhong Ding, Arya Rathee, and Wan Du. CLUE: Safe Model-Based RL HVAC Control Using Epistemic Uncertainty Estimation. In *ACM BuildSys*, 2023.
[66] Zhiyu An, Xianzhong Ding, and Wan Du. Go Beyond Black-box Policies: Rethinking the Design of Learning Agent for Interpretable and Verifiable HVAC Control. *arXiv preprint arXiv:2403.00172*, 2024.
[67] Ningning Hou, Xianjin Xia, Yifeng Wang, and Yuanqing Zheng. One shot for all: Quick and accurate data aggregation for LPWANs. *IEEE/ACM Transactions on Networking*, 2024.
[68] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. CloakLoRa: A Covert Channel over LoRa PHY. *IEEE/ACM Transactions on Networking*, 2022.
[69] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-Level localization with a single WiFi access point. In *USENIX NSDI*, 2016.
[70] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. Deep learning based wireless localization for indoor navigation. In *ACM MobiCom*, 2020.
[71] Jait Purohit, Xuyu Wang, Shiwen Mao, Xiaoyan Sun, and Chao Yang. Fingerprinting-based indoor and outdoor localization with LoRa and deep learning. In *IEEE GLOBECOM*, 2020.
[72] Hongxu Zhu, Kim-Fung Tsang, Yucheng Liu, Yang Wei, Hao Wang, Chung Kit Wu, and Hao Ran Chi. Extreme RSS based indoor localization for LoRaWAN with boundary autocorrelation. *IEEE Transactions on Industrial Informatics*, 17(7):4458–4468, 2020.
[73] Yidong Ren, Puyu Cai, Jinyan Jiang, Jialuo Du, and Zhichao Cao. Prism: High-throughput LoRa backscatter with non-linear chirps. In *IEEE INFOCOM*, 2023.
[74] Caitao Zhan, Himanshu Gupta, Arani Bhattacharya, and Mohammad Ghaderibaneh. Efficient localization of multiple intruders in shared spectrum system. In *ACM/IEEE IPSN*, 2020.
[75] Caitao Zhan, Mohammad Ghaderibaneh, Pranjal Sahu, and Himanshu Gupta. Deepmtl: Deep learning based multiple transmitter localization. In *IEEE WoWMoM*, 2021.
[76] Caitao Zhan, Mohammad Ghaderibaneh, Pranjal Sahu, and Himanshu Gupta. Deepmtl pro: Deep learning based multiple transmitter localization and power estimation. *Pervasive and Mobile Computing*, 82:101582, 2022.
[77] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *ACM MobiCom*, 2012.
[78] Zheng Yang, Chenshu Wu, and Yunhao Liu. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *ACM MobiCom*, 2012.
[79] Xiaopeng Zhao, Zhenlin An, Qingrui Pan, and Lei Yang. NeRF^2: Neural Radio-Frequency Radiance Fields. In *ACM MobiCom*, 2023.
[80] Tribhuvanesh Orekondy, Pratik Kumar, Shreya Kadambi, Hao Ye, Joseph Soriaga, and Arash Behboodi. WiNeRT: Towards Neural Ray Tracing for Wireless Channel Modelling and Differentiable Simulations. In *ICLR*, 2023.
[81] Xianzhong Ding, Le Chen, Murali Emani, Chunhua Liao, Pei-Hung Lin, Tristan Vanderbruggen, Zhen Xie, Alberto Cerpa, and Wan Du. HPC-GPT: Integrating Large Language Model for High-Performance Computing. In *ACM SC-W*, 2023.
[82] Yifei Xu, Yuning Chen, Xumiao Zhang, Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu, Wan Du, Zhuoqing Mao, Ennan Zhai, et al. CloudEval-YAML: A Practical Benchmark for Cloud Configuration Generation. *arXiv preprint arXiv:2401.06786*, 2023.
[83] Huatao Xu, Liying Han, Qirui Yang, Mo Li, and Mani Srivastava. Penetrative ai: Making llms comprehend the physical world. In *ACM HotMobile*, 2024.