Generative Diffusion Model-Assisted Efficient Fingerprinting for In-Orchard Localization

Kang Yang, Yuning Chen, and Wan Du *Member, IEEE*

***Abstract*—Precise robot localization at the tree level is essential for smart agriculture applications such as precision disease management and targeted nutrient distribution. Existing methods fail to achieve the required accuracy. We propose *OrchLoc*, a fingerprinting-based localization solution that achieves tree-level precision using a single Long Range (LoRa) gateway. Our approach utilizes channel state information (CSI) across eight channels as a localization fingerprint. To minimize labor-intensive site surveys for fingerprint database construction and maintenance, we develop a CSI generative model (CGM) that learns the relationship between CSI vectors and their corresponding locations. The CGM is fine-tuned using CSI data from static agricultural LoRa sensor nodes, enabling continuous fingerprint database updates. Extensive experiments in two orchards demonstrate that *OrchLoc* effectively achieves accurate tree-level localization with minimal overhead, improving robot navigation.**

***Index Terms*—In-Orchard Localization, LoRaWAN, Fingerprinting, Generative Diffusion Model**

I. Introduction

Robots have become essential in precision agriculture for tasks such as pruning, harvesting, and spraying [1], [2], [3], [4]. These robots operate at the tree level, performing targeted actions to optimize resource use. For instance, proactive health assessments on individual trees enable timely interventions, preventing disease spread [5]. In viticulture, precise nutrient and pesticide delivery tailored to each vine reduces resource consumption while enhancing crop quality [6].

While meter-level localization is readily achievable in urban environments [7], [8], achieving tree-level localization in orchards remains challenging. Conventional approaches, including wheel encoders [9], Simultaneous Localization and Mapping (SLAM) [10], [11], and Global Positioning System/Inertial Navigation System (GPS/INS) [12], lack the precision to differentiate individual trees. Wheel encoders are prone to slipping in muddy conditions [13], while camera-based SLAM is affected by variable lighting, obstructions, high power consumption, and uneven terrain [14]. Light Detection and Ranging (LIDAR)-based SLAM provides high accuracy but is prohibitively expensive (over $10,000) [15], [16], [17]. GPS/INS systems [12] offer a cost-effective and energy-efficient solution by using GPS to correct inertial navigation drift [12]. However, even advanced methods like Real-Time Kinematic (RTK) GPS suffer from signal obstruction due to tree canopies [12], [14]. Our experiments in a pistachio orchard reveal an average GPS error of 7.9 m, and state-of-the-art GPS/INS methods [12] result in a tracking error of 9.1 m. Such errors make precise navigation infeasible, given that the typical tree spacing is less than 4.9 m.

Long Range (LoRa) networks, known for their wide coverage and low power consumption, have been deployed in orchards for applications such as smart irrigation [18], [19] and pest monitoring [20], [21]. This paper explores leveraging existing LoRa infrastructure for in-orchard localization. A robot's position is determined based on the packets it transmits to a LoRa gateway, and integrating LoRa-based localization with inertial navigation enhances overall positioning accuracy.

Several algorithms have been proposed for LoRa localization [22], [23], [24], but they are ineffective in orchards. These methods rely on multiple gateways receiving the same packet, whereas orchard deployments often use sparse gateway placement to reduce costs, leaving many sensor nodes within range of only a single gateway. Moreover, they assume a direct propagation path between nodes and gateways. In orchards, however, gateways are mounted on high poles for broader coverage, while nodes are positioned under tree canopies for environmental monitoring. As a result, a direct propagation path between the transmitter and receiver is rare due to obstructions from leaves and trunks, limiting the effectiveness of existing LoRa-based localization approaches.

This paper presents *OrchLoc*, a LoRa-based localization system for orchards that operates with a single gateway. We introduce a novel LoRa signal fingerprint, the channel state information (CSI), derived from the amplitude and phase spectra of signals received across eight channels by a dual-antenna gateway. Unlike Wi-Fi CSI fingerprinting methods, which separately process amplitude [25] and phase [26], [27], we design a location classifier with complex-valued Fully-Connected (FC) layers to process CSI fingerprints holistically. Experimental results demonstrate that our classifier improves precision by 20.3% and 46.7% compared to amplitude-only and phase-only approaches, respectively.

As a fingerprint-based localization system [28], *OrchLoc* faces two key challenges: the labor-intensive fingerprint site survey in large orchards and the aging of fingerprints. To address these challenges, this paper introduces two observations that enable efficient fingerprint database building and updating:

- *(i) Media Homogeneity:* Locations with sensor nodes deployed for precision agriculture can periodically collect CSI fingerprints to update their fingerprints over time. However, updating fingerprints at locations without sensor nodes remains a challenge. While CSI fingerprints vary across locations, LoRa signals primarily propagate through three media: air, foliage, and ground. Our experiments reveal that each medium exhibits consistent intrinsic shadowing effects, which can be

Kang Yang, Yuning Chen, and Wan Du are with the Department of Computer Science and Engineering, University of California, Merced, CA 95340 USA. E-mail: {kyang73, ychen372, wdu3}@ucmerced.edu.

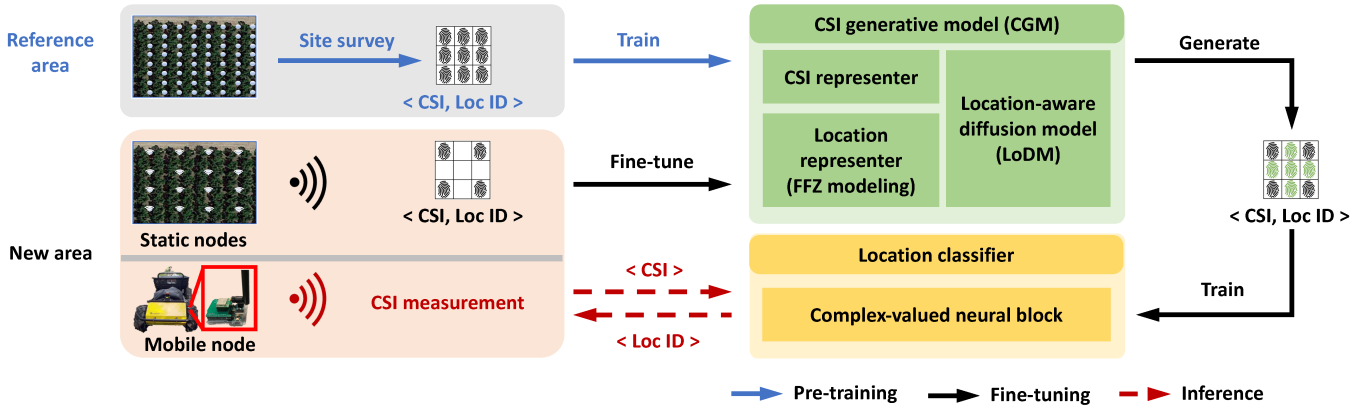


Fig. 1. Illustration of the workflow of *OrchLoc*. The combination of pre-training and fine-tuning stages is referred to as the turbo-training scheme.

learned from a subset of locations and generalized to others.

• *(ii) Spatial Homogeneity:* The uniform tree shapes and structured layouts in modern orchards allow for even LoRa gateway distribution, segmenting the orchard into distinct areas, each covered by a specific gateway. Each gateway maintains its own fingerprint database, and our experiments demonstrate high similarity in CSI fingerprints across areas, allowing one area's database to generate fingerprints for others.

Inspired by advances in generative models [29], [30], we introduce a *CSI Generative Model (CGM)* to generate CSI fingerprints based on location IDs [1]. As depicted in Figure 1, leveraging the CGM, we propose a turbo-training scheme for efficient fingerprint database construction and updates in orchards. First, we pre-train the CGM using fingerprints from a reference area, taking advantage of its representative features due to *Spatial Homogeneity*. Next, the pre-trained CGM is fine-tuned with sensor node data from a new area, capturing local features to enhance adaptability. By treating the CGM as a "signal propagation model," *Media Homogeneity* ensures that the fine-tuned CGM can generate CSI fingerprints for locations without sensor nodes. These generated fingerprints enable classifier updates without manual fingerprint measurements.

CGM employs a location-aware diffusion model (LoDM) to generate CSI data for each location. It leverages a complex-valued U-Net framework and attention layers to learn the latent relationship between CSI data and location IDs. However, directly inputting raw CSI data and location IDs into LoDM limits its modeling capability, as location IDs alone lack shadowing-specific information crucial for generating location-dependent data. Additionally, low-dimensional CSI vectors increase the risk of overfitting. To address these challenges, the CGM integrates specialized representers for both CSI and location IDs. A complex-valued autoencoder-based CSI representer transforms CSI data into high-dimensional vectors that capture key latent features. Meanwhile, the location representer models the First Fresnel Zone (FFZ), an ellipsoidal region surrounding the direct path between transmitter and receiver that contains most of the transmitted signal energy [31]. It quantifies the proportions of foliage, air, and ground within the FFZ and incorporates the communication distance and direction to construct a detailed FFZ vector. This approach provides a more robust, physically informed representation than using location IDs alone.

We collect thirteen CSI fingerprint databases over thirteen data collection rounds spanning four months in a pistachio orchard area and gather ten databases from ten areas across two pistachio orchards. The turbo-training scheme consistently maintains an average precision of 96.3% and recall of 97.6% over four weeks. Across the ten areas, *OrchLoc* achieves an average precision of 89.6% and recall of 91.8%, with a localization error of just 1.2 m. Furthermore, replacing GPS with *OrchLoc* for robot navigation reduces navigation error by 61.3% using the Neural Kalman Filter algorithm [12].

In summary, this paper makes the following contributions:

• We develop *OrchLoc*, achieving tree-level in-orchard localization accuracy using a single LoRa gateway.

• We enhance CGM to generate high-quality CSI fingerprints by incorporating a location-aware diffusion model, a CSI representer, and an FFZ-based location representer.

• Extensive experiments validate the system's localization accuracy and its effectiveness in improving robot navigation.

II. Preliminary

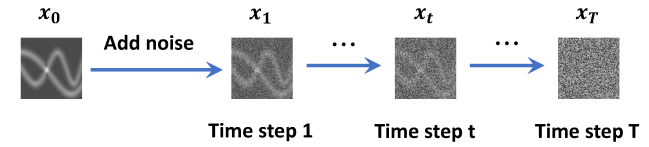
A. LoRa Primer

LoRa employs Chirp Spread Spectrum (CSS) modulation to transmit signals [32], [33]. In CSS, a chirp is a signal whose frequency continuously increases or decreases across a bandwidth, referred to as up-chirps and down-chirps, respectively. A LoRa packet is modulated into a sequence of chirps, beginning with preambles. These preambles typically consist of a fixed number of standard up-chirps, such as eight in LoRa Wide Area Network (LoRaWAN) [34]. The frequency of these up-chirps starts at the minimum value and progressively sweeps to the maximum frequency within the bandwidth. This structured preamble enables the measurement of CSI by comparing received chirps with a reference up-chirp.

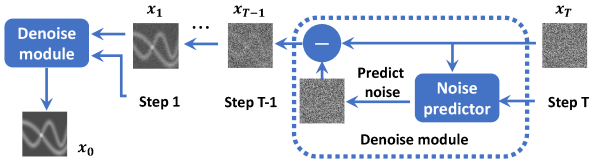
B. Diffusion Models

Diffusion models are a class of probabilistic generative models that reconstruct data by reversing a noise corruption process [35]. This work focuses on the Denoising Diffusion Probabilistic Model (DDPM) [29], illustrated in Figure 2, which consists of forward and reverse Markov chains.

[1]The terms "location ID," "tree ID," and "location (tree)" are used interchangeably to denote the unique identifier for each tree in the orchard.



(a) Forward Markov chain



(b) Reverse Markov chain

Fig. 2. Illustration of the standard diffusion models.

The forward Markov chain, depicted in Figure 2(a), progressively transforms data into an isotropic Gaussian distribution $\mathcal{N}(0, I)$ by incrementally adding Gaussian noise with zero mean and variance β_t over time steps $t \in [1, T]$. Here, T denotes the total number of diffusion steps, and β_t increases with t. The identity matrix I represents independent and identically distributed noise across all dimensions.

The reverse chain, illustrated in Figure 2(b), reverses this transformation using a denoising module with a noise predictor parameterized by deep neural networks. The predictor estimates the noise in the corrupted data $\mathbf{x}_t$ at step t and subtracts it to obtain $\mathbf{x}_{t-1}$. This iterative denoising process continues from T to 1, ultimately reconstructing the original data $\mathbf{x}_0$. To generate new samples, a random vector is drawn from a Gaussian distribution and denoised via ancestral sampling through the reverse Markov chain [36].

The success of this sampling process relies on accurately training the reverse Markov chain to approximate the time-reversed forward chain. This involves optimizing θ so that the joint distribution of the reverse process, $p_\theta(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T)$, closely matches that of the forward process, $q(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T)$. To achieve this, the noise predictor is trained to estimate the noise added during the forward process, taking as input: (i) the time step t, indicating the noise level, and (ii) the corrupted sample $\mathbf{x}_t$. The network outputs the predicted noise $\hat{\epsilon}_\theta$, enabling accurate iterative denoising to recover $\mathbf{x}_0$. By precisely predicting the noise, the model supports high-quality sample generation from Gaussian noise.

III. Motivation

This section examines the impact of GPS accuracy on the navigation performance of the GPS/INS system.

We use the public dataset [12], where a robot navigates a farm over a total trajectory length of 2.0 km. The robot is equipped with a Bosch BNO055 to record INS data. An OptiTrack 13W-P MoCa system captures the robot's initial position and continuous velocity, providing ground truth for trajectory evaluation. To simulate orchard GPS accuracy (7.9 m), we introduce Gaussian noise with a mean of 7.9 m and a standard deviation of 1.0 to the real coordinates.

For GPS/INS navigation system, the Extended Kalman Filter (EKF) [37] fuses GPS and INS data, using GPS to correct INS drift. The Neural Kalman Filter (Neurl-KF) [12] predicts

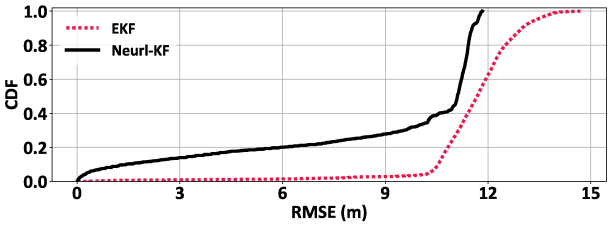


Fig. 3. Robot trajectory error is measured as the root-mean-square error (RMSE) between the predicted and ground truth trajectories.

the robot's velocity and position from INS data via a neural network, refining estimates through Kalman filtering with GPS updates. As shown in Figure 3, the navigation error, measured by the root-mean-square error (RMSE) between predicted and true trajectories, averages 11.6 m with EKF and 9.1 m with Neurl-KF. These large errors result from the system's reliance on GPS for correcting cumulative INS drift. When GPS accuracy degrades, as in orchard environments, the ability to compensate for INS drift is significantly reduced, leading to substantial navigation errors. Such inaccuracy makes GPS-based robot navigation impractical in orchards, where tree spacing is typically around 4.9 m (*e.g.*, almond orchards). This paper proposes an alternative to GPS for correcting INS drift.

IV. Localization using *OrchLoc*

We begin by introducing CSI for LoRa signals and designing a location classifier that leverages complex-valued FC blocks to process complex-valued data, enabling CSI fingerprint-based localization. We then identify two key challenges in fingerprint-based in-orchard localization and present two observations that offer solutions in orchard environments.

A. CSI Fingerprint

CSI characterizes how signals propagate from sender to receiver across different carrier frequencies and multiple paths [38]. In Wi-Fi systems, CSI is obtained by transmitting known pilot symbols at specific subcarriers, followed by applying the Fast Fourier Transform (FFT) to convert the received time-domain signal into the frequency domain. Estimation algorithms, such as Least Squares, are then used to compute the channel frequency response at the pilot subcarriers. However, this method is not applicable to LoRa, as CSS modulation operates without dedicated pilot symbols.

1) Extracting CSI for LoRa: To extract stable CSI data for LoRa, we apply a series of processing steps: de-chirping, FFT, preamble calibration, phase rotation compensation for amplitude calculation, and phase estimation based on the phase difference between two antennas. Upon receiving a packet, the CSI, represented as a complex number capturing the channel frequency response, is obtained by comparing the amplitude and phase spectra of the packet preambles with those of a standard up-chirp. The preambles are well-calibrated, as described in [39], for subsequent calculations.

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 CSI amplitude is determined as the ratio of the peak height of the

received preamble to that of the standard up-chirp. To mitigate 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 $\phi \in [0, 2\pi]$.

Phase: The phase of the peak is first computed from two antennas. Next, the phase difference between the two antennas is derived to obtain more stable phase data over time [39], [40].

These operations ensure the extraction of stable CSI data while minimizing potential interference from other wireless systems operating in the same unlicensed frequency band. Notably, we do not account for cell edge interference from adjacent gateways, as in the United States, LoRa gateways are configured to operate on different frequency channels [41], preventing signal interference between neighboring gateways.

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

Fingerprint Database: CSI fingerprints are collected at M = 64 locations (trees) within a gateway's coverage area, with each location positioned 2.0 m to the right of a tree, as shown in Figure 14(b). At each location, 160 packets are recorded. Each set of eight consecutive packets, transmitted across the eight channels, forms a CSI vector, yielding 20 CSI vectors per location. A data augmentation technique [42] is applied to generate an additional 80 CSI vectors per location by introducing Gaussian noise to the initial 20 CSI vectors. This noise is sampled from a zero-mean Gaussian distribution, with the standard deviation for each dimension determined by the standard deviation of the collected 20 CSI vectors.

2) Spatial Resolution: We utilize the Multiple Signal Classification (MUSIC) algorithm [43] and the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique [44] to analyze the spatial resolution of CSI fingerprints from both physical and representation learning perspectives.

The MUSIC algorithm is applied to LoRa samples collected from two antennas to assess signal strength at various angles. This process is repeated across eight channels to generate spatial spectra for each location, with each location positioned 2.0 m to the right of a tree. Figure 4 illustrates distinct spatial spectra for two neighboring locations, corresponding to two adjacent trees, indicating differences in signal arrival angles at the gateway. This distinction confirms that LoRa signals can effectively differentiate adjacent locations in orchards.

Unlike traditional methods that use time-of-arrival (TOA) or direction-of-arrival (DOA) as location-specific features [45], [46], which explicitly measure signal travel time or arrival direction, CSI captures complex interactions between transmitted signals and surrounding objects. Each CSI vector encodes a spatial signature, incorporating distortions caused by reflections, diffractions, and scattering. Since these distortions depend on the spatial geometry of the transmitter, receiver, and environment, CSI inherently embeds location-specific features.

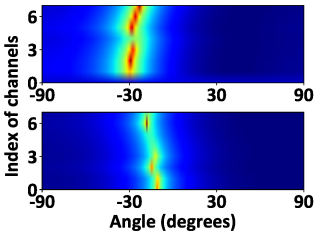


Fig. 4. Spatial spectra of two adjacent locations (trees) in an area.

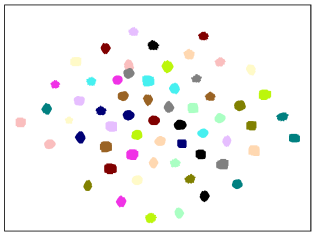


Fig. 5. Representation of all locations in an area using t-SNE [44].

Furthermore, Figure 5 presents the results of applying t-SNE [44], a widely adopted technique that reduces high-dimensional data to a two-dimensional space while preserving the pairwise similarity structure of the original data, to the CSI fingerprints across all locations. Markers of different colors and shapes represent distinct locations, and CSI data from the same locations exhibit a clear clustering effect, confirming the high spatial resolution of CSI fingerprints.

B. Complex-Valued Location Classifier

Flattening the CSI vector into a one-dimensional real-valued format enables traditional methods such as K-Nearest Neighbors (KNN) to be applied for localization. However, this transformation can degrade the spatial features of the CSI vector, compromising interdependencies among its elements. To mitigate this, we introduce a complex-valued FC block that effectively integrates both amplitude and phase information.

1) Complex-Valued FC Layer: The complex-valued FC layer consists of two real-valued neurons that separately process the real and imaginary parts of the input. If the input is represented as $\mathbf{c} = \mathbf{u} + i \cdot \mathbf{v}$, where $\mathbf{u}$ and $\mathbf{v}$ denote the real and imaginary components, the transformation is defined as:

$$\mathbf{u}' = \sigma\left(\mathrm{Re}\left(\mathbf{w} \cdot \mathbf{c} + \mathbf{b}\right)\right), \quad \mathbf{v}' = \sigma\left(\mathrm{Im}\left(\mathbf{w} \cdot \mathbf{c} + \mathbf{b}\right)\right) \tag{1}$$

where $\mathbf{w} = \mathbf{w_u} + i \cdot \mathbf{w_v}$ is the weight matrix, $\mathbf{b} = \mathbf{b_u} + i \cdot \mathbf{b_v}$ is the bias term, and σ is the ReLU activation function.

2) Transition to Real-Valued FC Layer: We employ two complex-valued FC layers. To connect the complex-valued FC layers with real-valued FC layers, we compute the absolute values of the outputs from the second complex-valued FC layer and use them as input to the real-valued layer. After transitioning to real-valued representations, the extracted features pass through two additional real-valued FC layers, each followed by a ReLU activation function. The final real-valued FC layer consists of M neurons, corresponding to the number of locations (trees) within a gateway's coverage area. The output is processed by a softmax function, transforming the logits into a probability distribution over M locations:

$$p_m = \frac{\exp(z_m)}{\sum_{j=1}^{M} \exp(z_j)} \tag{2}$$

where z_m is the logit output for location m.

3) Loss Function: To train the location classifier, we employ the cross-entropy loss function [47]:

$$\mathcal{L} = -\sum_{m=1}^{M} y_m \log p_m \tag{3}$$

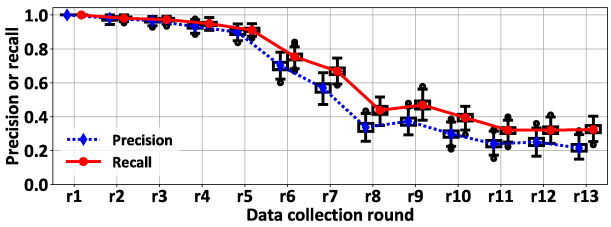


Fig. 6. Localization accuracy across thirteen data collection rounds.

where y_m is a one-hot encoded vector representing the true location ID. This loss function ensures that the predicted probability distribution aligns with the ground truth labels. By minimizing this loss, the model learns to classify a CSI vector into its corresponding location (tree) ID.

C. Performance and Challenges

Over four months, we conduct thirteen rounds ($r1$ to $r13$) of CSI measurements in an area with $M = 64$ locations (trees), referred to as Area A in Figure 14(a). In each round, we collect a CSI fingerprint database for all locations, resulting in thirteen distinct datasets recorded at different times: $r1$ (07/06, first day AM), $r2$ (07/06, first day PM), $r3$ (07/07, second day), $r4$ (07/08, third day), $r5$ (07/15, tenth day), $r6$ (07/22, seventeenth day), and so on. Each database measurement round, covering all $M = 64$ locations, lasts approximately four hours. Environmental parameters observed during each round are listed in Table I. Each database is split into training and testing sets with a 7:3 ratio. The location classifier, trained solely on the $r1$ training set, is then evaluated on testing sets from $r1$ to $r13$ without retraining on rounds $r2$ to $r13$.

Figure 6 illustrates the consistently high accuracy of our classifier across the first five rounds ($r1$-$r5$), with average precision and recall stabilizing above 89.8% and 91.1%, respectively. This stability is attributed to three key factors: *First,* the integration of techniques proposed in Section IV-A1, including de-chirping, FFT, phase rotation compensation for amplitude calculation, and phase difference estimation using two antennas, enhances CSI stability [39], [40]. *Second,* environmental conditions remain relatively stable during this period, as shown in Table I. Specifically, an average temperature increase of only 3.1°C from $r1$ to $r2$-$r5$ suggests that the fingerprint database remains robust when temperature fluctuations are within this threshold. *Finally,* orchards typically experience minimal external disturbances such as human activity or vehicular traffic, further contributing to stability.

Aging Problem: Figure 6 also reveals a noticeable decline in classifier accuracy starting from round $r6$, with precision and recall dropping to 56.8% and 66.5%, respectively. This degradation is primarily caused by a shift in fingerprint distribution due to changing environmental conditions from round $r6$ onward compared to round $r1$. In particular, Table I highlights a substantial temperature increase of 10.5°C from $r1$ to $r6$ and $r7$. These results indicate that periodic updates to the CSI fingerprint database are necessary when temperature variations exceed 3.1°C to maintain classifier accuracy.

Vast Orchard: In a 100-acre orchard with row and column spacing of 6.7 m and 4.9 m, and a LoRa communication range of 120 m, the orchard is divided into 29 areas, each containing 19×26 trees (locations) [48]. Allocating three minutes per location for data collection results in a total survey time of 29.8 days. The requirement for periodic database updates further complicates this process, making site surveys impractical.

TABLE I
ENVIRONMENTAL DYNAMICS STATISTICS ACROSS THIRTEEN DATA COLLECTION ROUNDS OVER FOUR MONTHS.

Date	Hum (%)	Tem (°C)	Pre (mm)	Win (m/s)
07/06 (r1)	39.0 ± 8.5	26.7 ± 2.6	0.0 ± 0.0	2.1 ± 0.1
07/06 (r2)	34.3 ± 3.5	31.0 ± 0.9	0.0 ± 0.0	1.9 ± 0.3
07/07 (r3)	42.3 ± 3.1	29.9 ± 0.4	0.0 ± 0.0	2.1 ± 0.4
07/08 (r4)	43.7 ± 1.5	27.8 ± 1.5	0.0 ± 0.0	2.1 ± 0.3
07/15 (r5)	56.3 ± 18.6	31.2 ± 5.8	0.0 ± 0.0	0.8 ± 0.4
07/22 (r6)	39.3 ± 11.5	36.8 ± 2.7	0.0 ± 0.0	1.0 ± 0.2
07/29 (r7)	24.7 ± 3.5	35.5 ± 0.7	0.0 ± 0.0	2.0 ± 0.0
11/18 (r8)	79.0 ± 9.0	16.9 ± 1.3	0.2 ± 0.3	2.2 ± 1.3
11/19 (r9)	47.3 ± 0.6	15.8 ± 0.2	0.0 ± 0.0	5.1 ± 0.3
11/25 (r10)	46.0 ± 2.0	15.3 ± 0.9	0.0 ± 0.0	1.7 ± 0.1
12/09 (r11)	49.6 ± 2.1	14.3 ± 0.5	0.0 ± 0.0	1.4 ± 0.2
12/16 (r12)	47.6 ± 8.6	11.9 ± 3.1	0.0 ± 0.0	1.4 ± 0.4
12/23 (r13)	69.5 ± 3.3	15.1 ± 0.5	0.0 ± 0.0	1.1 ± 0.2

* Hum: Humidity, Tem: Temperature, Pre: Precipitation, Win: Wind

D. Two Key Observations

We identify two key observations in orchards that help reduce labor for building and updating the fingerprint database.

1) Media Homogeneity: In orchards, signals propagate through three primary media: air, foliage, and ground. Each medium introduces distinct shadowing effects on signal propagation, which remain consistent across different locations. An experiment focusing on Received Signal Strength Indicator (RSSI) estimation is conducted to illustrate this concept.

The Log-Normal Shadowing model ("Log") [31] is used to estimate the RSSI of received LoRa packets. This model employs the Path Loss Exponent (PLE) to characterize the shadowing effects of different media on signal propagation. Since signals in orchards traverse air, foliage, and ground, PLE is decomposed into three components [48]:

$$PLE = P_{air} \times \alpha + P_{foliage} \times \beta + P_{ground} \times \gamma \quad (4)$$

where P_{air}, $P_{foliage}$, and P_{ground} represent the proportions of air, foliage, and ground in the signal propagation zone. The parameters α, β, and γ denote the intrinsic PLE values for signals propagating through each respective medium.

We collect RSSI data at four orchard locations over four weeks. Data from the first week are used to fit the values of α, β, and γ. The Log model is then evaluated over the following three weeks in two scenarios: (i) without updating, and (ii) updating α, β, and γ using data from a single location.

Figure 7 presents the estimation error for both cases. In case 2, the error is reduced from 13.9 dB to 5.7 dB across all locations compared to case 1. This improvement is attributed to the media homogeneity in orchards, where all locations share the same α, β, and γ values. Thus, updating the Log model with data from just one location effectively enhances accuracy across the entire orchard. In contrast, urban environments contain diverse media, complicating PLE decomposition and reducing the effectiveness of this updating scheme.

Hence, in orchards, updating propagation models with data from a subset of locations improves accuracy across the area.

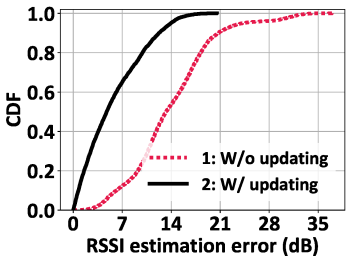


Fig. 7. Updating a propagation mode's parameters using a subset of locations.

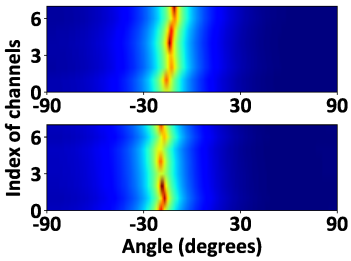


Fig. 8. Spatial spectra of the same location ID from two areas (PSNR: 25.1 dB).

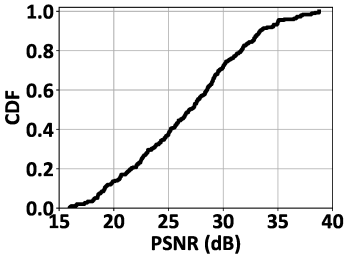


Fig. 9. Spatial spectra similarity across corresponding locations in six areas.

2) Spatial Homogeneity: In modern orchards, trees have uniform shapes and structured layouts. The orchard is divided into multiple areas, each with a uniform tree arrangement and covered by a LoRa gateway. Each gateway maintains a CSI fingerprint database for its coverage area. The similarity of these databases across areas, resulting from layout uniformity, is referred to as spatial homogeneity.

To quantify this similarity, we use the Peak Signal-to-Noise Ratio (PSNR) to compare the spatial spectra of identical location IDs across different areas. PSNR is defined as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\max(X)^2}{\text{MSE}(X, X')}\right) \tag{5}$$

where X represents the spatial spectra of a location ID from one area, and X' is the corresponding spectra from another area for the same location ID. $\max(X)^2$ denotes the maximum possible power of the spectra, while $\text{MSE}(X, X')$ represents the mean squared error (MSE) between the two spectra. Higher PSNR values indicate greater similarity between spectra, as PSNR is inversely related to MSE. Smaller spectral differences suggest structural consistency in CSI fingerprints.

As illustrated in Figure 8, the PSNR for the same location IDs across two areas reaches up to 25.1 dB, indicating strong spectral similarity. Additionally, Figure 9 presents the PSNR distribution of spatial spectra for corresponding location IDs across six areas, showing that 86.7% of the spectra have a PSNR above 20 dB. Similar spatial spectra correspond to comparable CSI fingerprints, as they reflect signals propagating through similar environments. These findings confirm that various areas within the orchard exhibit spatial homogeneity.

V. Workflow of *OrchLoc*

Figure 1 illustrates the workflow of *OrchLoc*, which efficiently builds and updates databases in orchards. *OrchLoc* operates in three stages: pre-training, fine-tuning, and inference.

• In pre-training, we designate one area as a reference and collect CSI fingerprints from all its locations to serve as an initial database. This database pre-trains the CSI Generative Model (CGM), which integrates a location-aware diffusion model with CSI and FFZ-based location representers, enabling the generation of CSI fingerprints for various locations.

• In fine-tuning, the process is designed for both building and updating the fingerprint database for a specific area:
(i) Building: Fingerprints from locations equipped with static sensor nodes are used to fine-tune the pre-trained CGM, adapting it to local variations. These static sensor nodes are already deployed in orchards for precision agriculture applications, such as soil moisture monitoring. The refined CGM generates fingerprints for locations without sensors, which are then combined with data from sensor-equipped locations to create a complete database. This database is subsequently used to train the area-specific complex-valued location classifier.
(ii) Updating: Instead of fine-tuning the initial pre-trained CGM, the most recent area-specific refined CGM is fine-tuned to account for temporal environmental changes.

• In inference, the trained location classifier estimates the robot's location given its current CSI measurement.

The pre-training and fine-tuning stages, collectively referred to as turbo-training, leverage two key observations in orchards: First, using fingerprints from all locations in a reference area exploits *spatial homogeneity*, as the database characteristics of this area mirror those of other areas. Second, the success of fine-tuning across all locations is due to *media homogeneity*. Although the CGM is more complex than the Log model, its parameters function similarly to Log's α, β, and γ, which represent the shadowing effects of different media on CSI fingerprints. Thus, the fine-tuned CGM parameters remain applicable across all locations within an area.

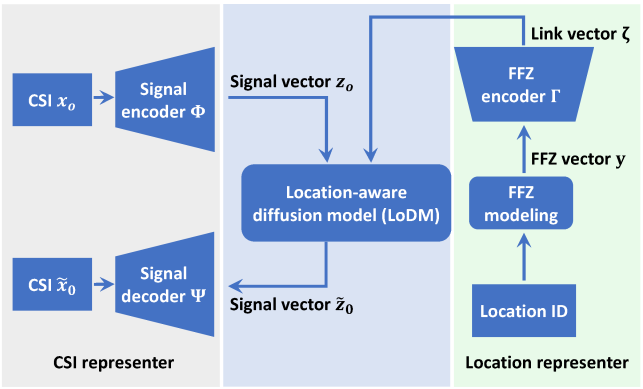


Fig. 10. Architecture of the CSI Generative Model (CGM).

VI. CSI Generative Model

Figure 10 outlines the three components of the *CGM*.

• (i) The location-aware diffusion model (LoDM), receiving input from CSI and location representers, learns the relationship between CSI fingerprints and location IDs.

• (ii) The CSI representer employs a complex-valued autoencoder, where the encoder Φ transforms low-dimensional CSI into a high-dimensional signal vector $\mathbf{z}_0$, and the decoder Ψ reconstructs the raw CSI $\tilde{\mathbf{x}}_0$ from the signal vector $\tilde{\mathbf{z}}_0$ generated by the LoDM module.

• (iii) For the location representer, the location ID is converted into an FFZ vector $\mathbf{y}$ via FFZ modeling, which is then processed by an FFZ encoder Γ into a link vector ζ.

Pairs of CSI fingerprints and their corresponding location IDs are fed into the CGM for training. To generate CSI fingerprints, the trained CGM takes a location ID and Gaussian noise matching the size of the CSI fingerprint as input.

A. CSI Representer

Raw CSI fingerprint inevitably contains measurement noise from environmental factors such as fluctuating atmospheric conditions and transient disturbances. Directly using noisy CSI data can mislead the localization model, causing it to interpret transient variations as meaningful features. Additionally, low-dimensional CSI representations often suffer from overfitting [49], where the model becomes overly specialized to specific data patterns, degrading its generalization capability to unseen CSI samples. To address these challenges, we adopt an autoencoder-based transformation to derive high-dimensional CSI representations that enhance robustness against noise and improve feature expressiveness.

Our autoencoder consists of an encoder Φ and a decoder Ψ, both composed of three complex-valued FC layers. The encoder progressively expand in dimension, while the decoder symmetrically contracts, ultimately reconstructing the raw CSI vector $\tilde{\mathbf{x}}_0$. To ensure non-linearity in feature extraction, a ReLU activation function follows each FC layer. This transformation serves two purposes:

Mapping low-dimensional CSI into a structured higher-dimensional space enables the encoder to filter out high-frequency noise while preserving essential localization features. The complex-valued FC layers jointly process amplitude and phase information, capturing richer spatial dependencies that raw CSI representations fail to encode explicitly. This approach is similar to kernel methods in Support Vector Machines (SVMs) [50], where low-dimensional data points are mapped into a higher-dimensional space for improved separability. Likewise, our autoencoder generates structured high-dimensional CSI features, enhancing localization accuracy.

To train the complex-valued autoencoder, we minimize the reconstruction error using the L_2 loss function:

$$\mathcal{L}_{L_2}(\mathbf{x}_0, \tilde{\mathbf{x}}_0) = \|\mathbf{x}_0 - \tilde{\mathbf{x}}_0\|_2^2 \tag{6}$$

where $\mathbf{x}_0$ represents the original CSI vector, and $\tilde{\mathbf{x}}_0$ denotes the reconstructed CSI vector from the decoder. The L_2 loss function computes the squared Euclidean distance between these two vectors, ensuring that the reconstructed CSI closely approximates the original input data.

B. Location Representer

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

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 has limitations, such as reduced training efficiency and the lack of physical location context

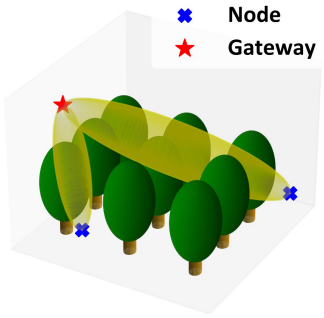


Fig. 11. Illustration of the First Frozen Zone (FFZ) modeling.

to guide the LoDM in learning CSI-location relationships. To address this, FFZ modeling is introduced. It transforms a location ID into a 7-element FFZ vector $\mathbf{y}$ that encapsulates physical factors influencing signal transmission.

The FFZ (Figure 11) represents a 3D ellipsoid region where most of the signal's energy is concentrated, with focal points aligned to the 3D coordinates of the node and gateway. In orchards, the FFZ consists of three media—air, foliage, and ground—each contributing distinct shadowing effects on the signal. To characterize the signal path, we calculate the proportions of these three media within the FFZ.

To achieve this, a 3D representation of the orchard is constructed using a Cartesian coordinate system, where the x- and y-axes align along and across orchard rows, and the z-axis points upward from the gateway's position on the ground. Uniform tree spacing determines tree positions on the x- and y-axes, while trees are modeled as cylinders (trunks) and ellipsoids (crowns), requiring only measurements of height and canopy width. Since trees exhibit consistent growth patterns within an orchard, modeling based on a single representative tree enables an efficient 3D representation

Using the 3D coordinates of the FFZ's focal points and its mathematical formulation [31], the FFZ is mapped onto the established coordinate system. Sampling points within the FFZ are analyzed to determine their 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 points within the FFZ.

Furthermore, both distance and direction between the node and gateway influence the received signal. To account for these factors, the normalized distance and 3D direction are incorporated, resulting in a 7-dimensional FFZ vector $\mathbf{y}$.

NOTE: Since localization is performed independently within each gateway's coverage region, transformations between different gateways' coordinate systems are unnecessary. The FFZ vector is computed solely within each gateway's local coordinate system. This design parallels the Log-Normal Shadowing model, where the RSSI depends on the relative distance between the node and the gateway rather than absolute global positions. By maintaining independent local coordinate systems, the model remains scalable and robust, simplifying localization while ensuring accurate location estimation within each gateway's coverage area.

2) FFZ Encoder: The FFZ vector $\mathbf{y}$ captures factors influencing the CSI vector, providing the LoDM with physical location information to learn the CSI-location relationship.

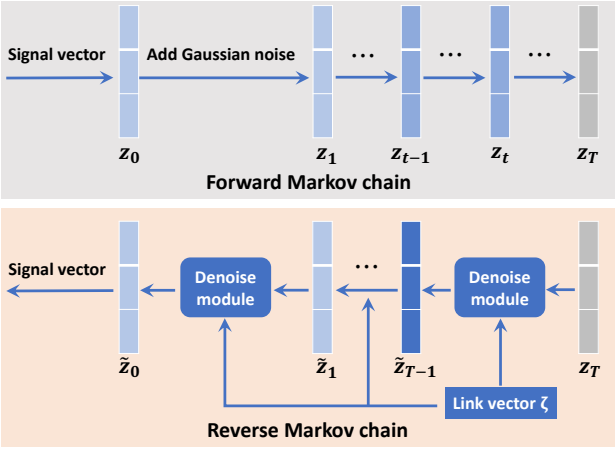


Fig. 12. Architecture of the Location-Aware Diffusion Model (LoDM).

However, due to uniform orchard layouts, the same FFZ vectors are generated for identical location IDs across different areas, limiting the LoDM's ability to adapt to new areas. 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. By refining the FFZ encoder during LoDM fine-tuning, the model adapts to local geometrical variations in new areas.

C. Location-Aware Diffusion Model

We develop the LoDM to generate CSI fingerprints across various locations, inspired by the DDPM [29].

1) Overview: Figure 12 illustrates the architecture of our LoDM, which consists of two Markov chains: forward and reverse. In the forward chain, unlike DDPM, which introduces noise to raw data, we apply noise to signal vectors $\mathbf{z}_0$ (high-order representations of raw CSI vectors $\mathbf{x}_0$), generating a sequence of noisy signal vectors $\mathbf{z}_t$. This is achieved by incrementally adding Gaussian noise to $\mathbf{z}_0$ at each step t.

In the reverse chain, the key task is to accurately remove noise from the noisy signal vector via the denoise module. Considering location effects on the signal, the module takes both the noisy signal vector $\mathbf{z}_t$ and the link vector ζ as inputs 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 $\tilde{\mathbf{z}}_t$ produces $\tilde{\mathbf{z}}_{t-1}$, which serves as input for the next denoising step at $t-1$. This iterative process continues from T to 1, ultimately recovering the initial signal vector $\tilde{\mathbf{z}}_0$.

Generating a signal vector requires a link vector ζ and a random vector $\mathbf{z}_T$ sampled from a Gaussian distribution, followed by ancestral sampling through the reverse chain [36].

2) Denoise Module: Figure 13 illustrates the denoise module, which features 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 the time step t and link vector ζ. The module then subtracts the estimated noise from $\tilde{\mathbf{z}}_t$, generating noisy signal vector $\tilde{\mathbf{z}}_{t-1}$ for the next time step $t-1$.

Complex-Valued U-Net Backbone: The U-Net backbone, shown in Figure 13, consists of contracting and expansive paths. The contracting path (right trapezoid) compresses the signal vector to extract key features and identify global trends. Conversely, the expansive path (left trapezoid) reconstructs the data, capturing fine details and preserving spatial correlations. This dual architecture enables the model to process the complex dimensions of the signal vector effectively.

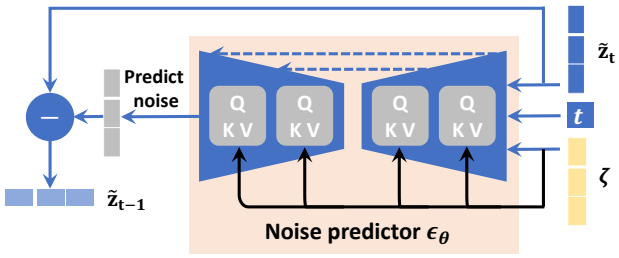


Fig. 13. Architecture of the denoise module with noise predictor.

In detail, the contracting path includes downsampling layers with 1D convolutions (increasing channel count), ReLU activations, residual layers (dotted lines), and a complex-valued FC layer for dimensionality reduction. The expansive path consists of upsampling layers with 1D convolutions (decreasing channel count), ReLU activations, residual layers, and a complex-valued FC layer for dimensional expansion.

Embedding Layer: The time step t is transformed from an integer into a vector using Sinusoidal Position Embeddings [51]. This embedding vector is then added to both the input and intermediate layers of the predictor.

Attention Layer: The noise predictor integrates the link vector ζ via a cross-attention layer [51], [52], [53], enabling the model to focus on location-relevant features. This improves the model's ability to generate accurate signal vectors for each location. The link vector ζ is incorporated into both downsampling and upsampling layers of the U-Net backbone through cross-attention mechanisms:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V \qquad (7)$$

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

3) Summary: Our system differentiates location fingerprints through a structured combination of spatial encoding and deep learning mechanisms. The location representer extracts the link vector ζ, which encodes spatial dependencies across different locations. This vector is integrated at every sampling layer in the U-Net, ensuring spatial consistency throughout the generative process. This approach parallels conditional diffusion models, where conditioning information guides generation while preserving essential spatial details.

Attention mechanisms within the U-Net emphasize the most distinctive spatial features in CSI data. Since the CSI vector spans multiple channels and antennas, the attention layers selectively highlight spatial variations critical for location differentiation. By learning to focus on these fine-grained spatial patterns, CGM ensures that the generated CSI fingerprints accurately correspond to their respective locations.

D. Calibration

Our continuous turbo-training scheme updates the fingerprinting dataset regularly, ensuring that the localization classifier is retrained to maintain high accuracy over time. Turbo-training efficiently adapts the CGM in dynamic environments with small, incremental changes, eliminating the need for full retraining. However, if the classifier's accuracy drops significantly, it indicates that turbo-training alone is insufficient to refresh the fingerprinting database. In such cases, more extensive calibration is required.

A larger portion of location data—or even the entire dataset—may need to be recollected and used to retrain the CGM. This is particularly crucial in environments with significant changes, such as the seasonal transition from a full-leaf orchard to a no-leaf setting, where physical conditions alter drastically. Such shifts can render previously collected fingerprints obsolete, making turbo-training alone inadequate for maintaining classifier accuracy.

To address this, we combine calibration with turbo-training. Calibration enables the CGM to adapt to major environmental changes by retraining on a broader dataset, while turbo-training continuously manages smaller updates in stable conditions. This joint strategy ensures that the localization classifier remains accurate and reliable, even in environments subject to substantial and unpredictable variations.

E. Generalizability to Other Environments

While our system has potential applications in various environments, such as wild forests, its generalizability depends on two key factors: an accurate 3D environment model for location representation and the presence of media and spatial homogeneity. These factors are essential for constructing and updating the fingerprint database. However, this reliance on spatial homogeneity also limits *OrchLoc* 's applicability to environments with highly irregular layouts, such as forests, urban areas, or orchards with significant terrain variations. Its effectiveness may degrade in scenarios where trees are unevenly distributed, as the lack of uniform spacing disrupts the consistency of fingerprinting patterns.

Additionally, since our method does not incorporate cross-domain adaptation techniques, its direct deployment in orchards with different crop species, canopy densities, or planting patterns may require extensive recalibration. Forests, in particular, introduce greater complexity due to irregular tree placement, species variations, and differences in age and shape, posing significant challenges [54]. This variability complicates fingerprinting and increases labor intensity for dataset collection and model adaptation.

Future research could explore lightweight adaptation of *OrchLoc* to more complex environments, such as through transfer learning-based methods [55], [56], [57], [58]. Potential approaches include integrating domain adaptation techniques to recalibrate CSI fingerprints across different environments, employing meta-learning strategies for efficient adaptation, or incorporating additional environmental sensing modalities (*e.g.*, LiDAR or hyperspectral imaging) to enhance robustness in highly heterogeneous landscapes.

VII. Implementation

Hardware: LoRa nodes are custom-built using the SX1276 Radio [59] on Arduino Uno boards [60]. The bladeRF 2.0 Software Defined Radio (SDR) [61], connected to a Raspberry Pi 4, is used to receive LoRa samples, and is equipped with two antennas spaced 14.0 cm apart—less than half the wavelength of the LoRa signal. Although dual-antenna configurations are not standard in commercial LoRa gateways, our setup explores their feasibility for in-orchard localization [62], [63]. Collected samples are processed on a local computer with an Intel(R) Core(TM) i9-11900KF @ 3.50 GHz CPU, while training is accelerated by an NVIDIA GeForce RTX 3080 Ti GPU, though GPUs are not required for inference.

Training of Location Classifier: The input layer requires data in the format $\mathbf{X} = \{x_r, x_i\} \in \mathbb{R}^{l\times 2}$, where the first and second columns represent the real and imaginary parts of the inputs, respectively. Thus, we flatten the $(2, 8)$ CSI vector into a 16-element column vector ($l = 16$) and compute the real and imaginary components for each element. The output layer size corresponds to the number of locations M.

Training of CGM: The CSI representer shares the same input format as the classifier. Its encoder generates the signal vector $\mathbf{z}_0$ with dimensions $(2, 128)$. The location representer, using FFZ modeling, creates a 7-element FFZ vector $\mathbf{y}$. The FFZ encoder then outputs a link vector ζ of length 32. To mitigate overfitting, a dropout layer with a probability of 0.1 is incorporated 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 the LoDM model, Gaussian noise with a mean of zero and 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 represents the total number of time steps in the diffusion process. We set $T = 1000$. The noise predictor is trained to minimize the discrepancy between predicted and actual noise. Training continues until a predefined convergence criterion is met, either by reaching the maximum number of epochs or achieving a loss below a specified threshold. In our implementation, training runs for a maximum of 300 epochs or until the loss falls below 0.01. The chosen hyperparameters align with those recommended in [29].

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

$$\mathcal{L}_{LoDM} = \mathbb{E}_{\Phi(\mathbf{x}_0),\mathbf{y},t,\epsilon\sim\mathcal{N}(0,1)}\left[\left\|\epsilon - \epsilon_\theta\left(\mathbf{z}_t, t, \Gamma\left(\mathbf{y}\right)\right)\right\|_2^2\right] \quad (8)$$

where $\mathbf{z}_t$ is the noisy version of $\mathbf{z}_0 = \Phi(\mathbf{x}_0)$ at time step t, with $\Gamma(\mathbf{y})$ producing the link vector ζ. The time step t is sampled uniformly from $\{1, \ldots, T\}$, ϵ represents the true noise added to $\mathbf{z}_t$, and $\epsilon_\theta(\mathbf{z}_t, t, \Gamma(\mathbf{y}))$ is the predicted noise.

The noise predictor ϵ_θ and the FFZ encoder Γ are jointly optimized through this loss function. This co-optimization ensures that both components are trained simultaneously, enabling the system to predict noise while effectively encoding the environmental characteristics of the FFZ. By aligning their training, the model improves its ability to handle noisy data and capture unique signal propagation effects within the FFZ.

VIII. Evaluation

We evaluate *OrchLoc* by analyzing its temporal and spatial performance in Sections VIII-B and VIII-C. Benchmarking results are presented in Section VIII-D, followed by a case study on robot navigation in Section VIII-E. Section VIII-F examines the effectiveness of the proposed components, while Section VIII-G investigates factors influencing system performance. Finally, Section VIII-H evaluates *OrchLoc*'s overhead.

A. Experimental Setup

1) Datasets: Pistachio orchards, a major economic crop producing approximately 1.1 billion pounds in 2021 [64], are selected for our evaluation. Figure 14(a) illustrates our pistachio orchard testbed, which is divided into multiple areas, each containing $M = 64$ trees. Area A serves as the reference area. In this orchard, trees are arranged with a spacing of 4.9 m between columns and 6.6 m between rows, with an average height of 6.1 m and a canopy width of 2.3 m. Within each area, LoRa nodes are positioned adjacent to the 64 trees, each placed 2.0 m to the right of a tree. At each node location, LoRa nodes transmit a packet to the gateway. As shown in Figure 14(b), the nodes and gateway are installed at heights of 0.45 m and 10.0 m, respectively. The system is 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.

Temporal Dimension Dataset: Thirteen CSI fingerprint databases, referred to as $r1$ to $r13$, are collected in Area A over thirteen rounds spanning four months, as listed in Table I.

Spatial Dimension Dataset: Ten CSI fingerprint databases are collected from six areas (A, B, C, D, E, F) in one pistachio orchard, as shown in Figure 14(a), and four areas $(B2, C2, D2, E2)$ in another pistachio orchard.

Training and Testing Sets: To maintain balanced location labels, each database partitions fingerprints for each location into training and testing sets with a 7:3 ratio.

2) Baselines: We evaluate the performance of *OrchLoc* against various location classifiers.

• GPS: We use Google Maps in satellite mode with an IC 5941 GPS radio [65] to determine location IDs and compare them with actual physical locations.

• RSSI: RSSI data from eight channels is used for location identification via real-valued FC layers.

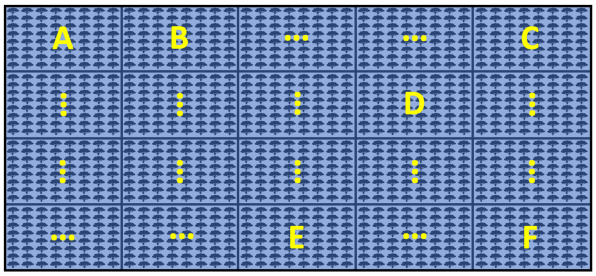
• AMP: Only the amplitude of CSI data is used for location identification via real-valued FC layers.

• PHA: Similarly, only the phase of CSI data is used for location identification via real-valued FC layers.

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

3) Performance Criteria: The location classifier maps a CSI vector to a specific location ID, making this a classification task. Using the testing dataset, we evaluate *OrchLoc* by computing the average precision and recall, which are then translated into localization error.

• Precision and recall are computed for each location: Precision is the ratio of correctly identified instances to all predictions for that location, while recall is 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 identifying tree locations.



(a) An orchard is divided into multiple areas.

(b) Setup in one area (one gateway and M locations).

Fig. 14. Illustration of the orchard testbed setup.

The importance of precision and recall depends on the application. For instance, in disease detection among trees, high recall is crucial to ensure all potentially diseased trees are flagged for further inspection, prioritizing complete identification over false positives. Conversely, other applications may require higher precision to minimize false identifications.

• Localization error is quantified as the Euclidean distance between the actual and predicted location IDs, considering the specific row and column spacing within the orchard. Each tree is assigned a unique location ID, with a reference point situated 2.0 m to the left of the tree's trunk.

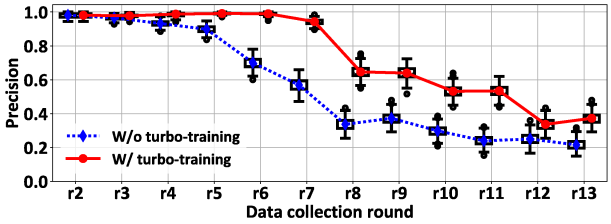
B. Temporal Dimension Performance

The performance of *OrchLoc* is influenced by various environmental dynamics, categorized into three types: (i) short-term weather changes (*e.g.*, temperature fluctuations), (ii) foliage density changes, (iii) long-term foliage shape changes.

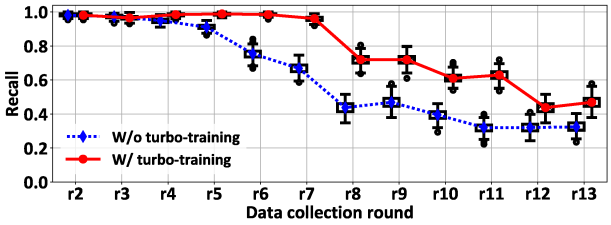
1) Dynamics (i) and (ii): Weather variations across thirteen rounds are detailed in Table I, while rapid pistachio growth in July suggests changes in foliage density [66]. To adapt to these dynamics from rounds $r2$ to $r13$, we randomly select 30% of location fingerprints from each round's database training set for updates using the turbo-training scheme, thereby refreshing the classifier. The updated classifier then identifies location IDs for the CSI measurements in each round's testing set.

Figure 15 illustrates that before round $r7$, our classifier achieves an average precision of 96.3% and recall of 97.6%, with a localization error of 0.4 m. This validates the effectiveness of turbo-training in managing dynamics i and ii, eliminating the need for manual CSI fingerprint collection.

However, by round $r8$, approximately three months after the initial data collection in round $r1$, Figure 15 shows a significant decline in performance despite turbo-training. To address this issue, the CGM model is recalibrated at round $r8$

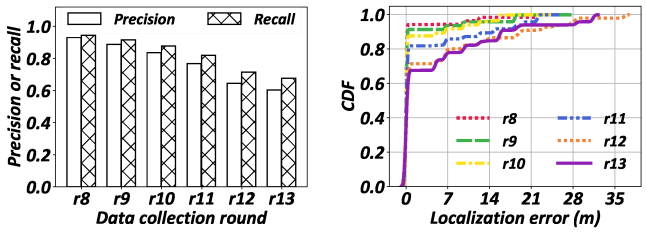


(a) Precision score



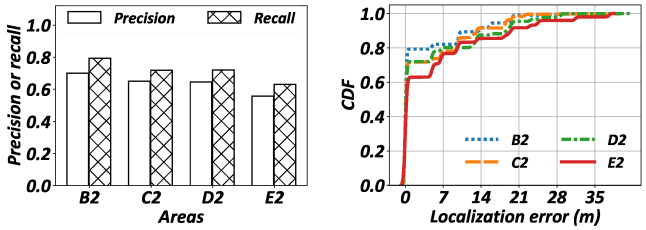
(b) Recall score

Fig. 15. Performance in Area A from data collection rounds $r2$ to $r13$, comparing results without (w/o) and with (w/) the turbo-training scheme.



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

Fig. 16. Performance following calibration at data collection round $r8$.



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

Fig. 17. Performance under long-term foliage shape changes (using the second pistachio orchard to simulate foliage shape changes in the first orchard).

by retraining it from scratch with data from all locations in that round. Turbo-training is then applied for rounds $r9$ to $r13$, and Figure 16 confirms that *OrchLoc* maintains high accuracy after this calibration. These results suggest that *OrchLoc* may require periodic calibration, such as once per season.

OrchLoc is beneficial during periods of dense foliage, as thick foliage can significantly hinder GPS signals. During seasons with sparse foliage, such as late autumn or winter, simpler GPS/INS systems may be sufficient for navigation due to clear GPS signals [12], rendering *OrchLoc* unnecessary.

The random selection of 30% of location fingerprints in each round is an intentional design choice that reflects real-world scenarios, where sensor placements can vary due to factors like terrain and canopy density. This approach prevents the classifier's performance from relying on specific static node placements and validates its robustness across diverse spatial subsets. Although the 30% subset changes in each round, the consistently high performance observed in spatial dimension experiments (Section VIII-C) shows that these gains are not merely the result of accumulated spatial coverage. Instead, they demonstrate the system's ability to generalize within each round, even when trained on a limited and randomly selected subset of locations. These experimental settings underscore the flexibility of *OrchLoc* in adapting to dynamic orchard conditions and reinforce its robustness for real-world deployments.

2) Dynamic (iii) - Long-Term Foliage Shape Changes: Foliage shape changes over the years due to trunk and branch growth. To evaluate this impact, we collect data in another pistachio orchard with slightly different traits than the first: row and column spacing of 6.6 m and 4.8 m, and average tree dimensions of 5.8 m in height and 2.1 m in width. We collect fingerprint databases from four areas ($B2$, $C2$, $D2$, and $E2$) in the second orchard. In each area, 30% of the locations' training sets are randomly selected for the turbo-training scheme, with the CGM pre-trained on data from Area A in the first orchard. The updated classifiers are then tested in these areas to evaluate their adaptability to long-term foliage shape changes.

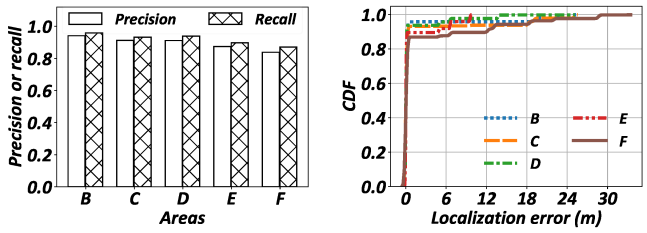
Figure 17 presents *OrchLoc* 's performance in the second orchard, with recall values of 79.3%, 71.9%, 72.1%, and 63.1%, and median localization errors of 2.8, 3.5, 4.1, and 5.3 m across four new areas. Despite some performance degradation due to orchard differences, the results remain promising, achieving an average precision of 63.8% and recall of 71.6% without manual data collection. This validates *OrchLoc*'s ability to reduce labor costs in training data collection for new orchards, demonstrating its efficiency and adaptability.

C. Spatial Dimension Performance

The turbo-training scheme is applied in five new areas ($B, C, D, E,$ and F) of the first orchard, with 30% of each area's training data randomly selected for turbo-training. This results in well-trained location classifiers for each area, which are then used for testing.

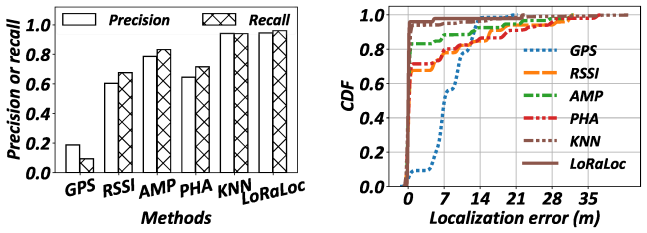
Figure 18 demonstrates *OrchLoc*'s effectiveness in new areas of the first orchard. We achieve precision values of 95.9%, 93.3%, 93.9%, 89.7%, and 87.1%, along with median localization errors of 0.5, 0.7, 0.8, 1.3, and 2.1 m in areas $B, C, D, E,$ and F, respectively. A notable trend is the performance degradation with increasing distances from area A, likely due to data distribution shifts caused by local variations. The accuracy in these areas is lower than in Figure 15, potentially due to both spatial and temporal diversity, as the data is collected in new areas and at different times relative to area A. To mitigate this, incorporating additional location data for turbo-training may be beneficial.

Potential Benefit: Consider a 100-acre orchard divided into 36 areas, each containing a grid of 30×25 trees (locations). Assuming that data collection at each location takes three minutes, collecting CSI data for a single area requires approximately 37.5 hours. Extrapolating this to the entire orchard, the conventional process would demand a staggering 1,350 hours. *OrchLoc* introduces a significant time-saving advantage. By collecting CSI data from only one representative area and leveraging trained models for the remaining areas, we



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

Fig. 18. Localization accuracy in the five areas of the first pistachio orchard.



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

Fig. 19. Localization accuracy in area A with six different location classifier.

exploit spatial similarity within the orchard. This reduces data collection time from 1,350 hours to just 37.5 hours, achieving a 36-fold reduction. Moreover, maintaining the database requires periodic site surveys. *OrchLoc* eliminates this need by leveraging already deployed agricultural static sensor nodes and media homogeneity to automatically update the fingerprint database. Not only does this approach streamline the process, but it also maintains the integrity and quality of localization.

D. Benchmark Study

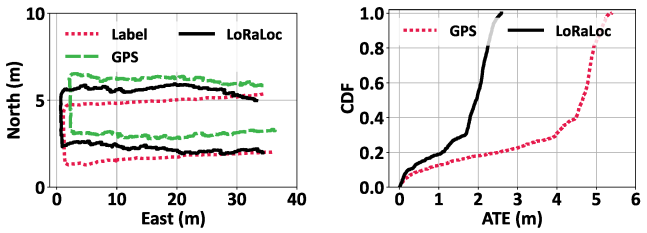
We assess five baselines on the dataset from Area A at round $r1$. The RSSI-based, amplitude-based, and phase-based classifiers use three FC layers. Amplitude and phase data are derived from CSI, while RSSI is computed using collected I and Q samples. For the KNN-based classifier, the location ID for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

Figure 19 shows that GPS has the lowest precision and recall, at 18.8% and 9.4%, respectively, with a median localization error of 7.9 m. This 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, improving precision by 56.6%, 20.3%, and 46.7%, and recall by 41.9%, 15.4%, and 34.3%, respectively. The median localization errors using RSSI, amplitude, phase, and CSI are 4.6, 2.7, 4.7, and 0.5 m, highlighting CSI's superiority in location identification. While amplitude alone is limited, incorporating phase information enriches location matching. Unlike simple FC layers, our complex-valued classifier fully utilizes CSI data, enhancing accuracy. KNN records precision and recall of 94.3% and 93.7%, compared to *OrchLoc*'s 96.3% and 97.6%. *OrchLoc*'s complex-valued FC block effectively captures the interaction between amplitude and phase in CSI, surpassing simple flattening approaches.

Additionally, the benchmark study supports the argument that classifier complexity is not the primary factor in localization accuracy. As demonstrated in Figure 19, even the simple KNN classifier achieves strong performance, with precision and recall values close to those of *OrchLoc*. This result indicates that the key determinant of localization accuracy is the model's ability to differentiate CSI data across locations rather than the complexity of the classifier itself.

Since our approach is classifier-agnostic, a FC layer serves as an effective backbone, balancing accuracy and computational efficiency while allowing for alternative classifiers if needed. Convolutional neural network (CNN)-based models [67] could also be integrated within our framework, potentially offering advantages in certain scenarios, depending on the structure of the CSI data and computational trade-offs.



(a) Trajectory reconstruction (b) Absolute trajectory error

Fig. 20. Comparison of robot navigation performance in an orchard: GPS *vs.* *OrchLoc* enhanced with INS data.

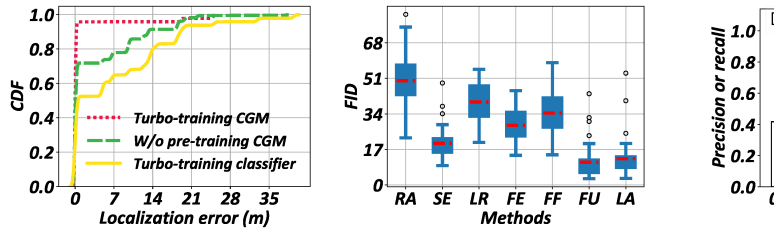
E. Case Study: Robot Navigation in Orchard

To evaluate *OrchLoc* for robot navigation in orchards, we utilize the public dataset [12], described in Section III, to simulate a robot's path. The process begins by aligning the robot's starting point with a specific tree in the orchard, with the initial speed set parallel to a tree row. The dataset follows a time-series format, containing INS data and ground truth coordinates for each timestamp. To integrate CSI data at each moment, we identify the nearest tree to the corresponding coordinate, retrieve all CSI measurements from that tree, and randomly select one to associate with the timestamp. GPS data is assigned in the same manner. The selected CSI vectors are then input into *OrchLoc* to infer location coordinates.

We combine GPS/INS and *OrchLoc*/INS by applying the state-of-the-art Neurl-KF [12]. Figure 20(a) visualizes the reconstructed trajectories, where *OrchLoc* demonstrates superior accuracy. As shown in Figure 20(b), *OrchLoc* yields lower navigation errors compared to GPS, averaging 1.2 m *vs.* 3.1 m. These results highlight *OrchLoc*'s potential to enhance navigation accuracy in GPS-challenged environments. The improved GPS accuracy in Figure 20 compared to Figure 3 is due to the simulation method incorporating tree location priors. Unlike the direct addition of Gaussian noise to real coordinates, tree locations remain fixed, reducing uncertainty.

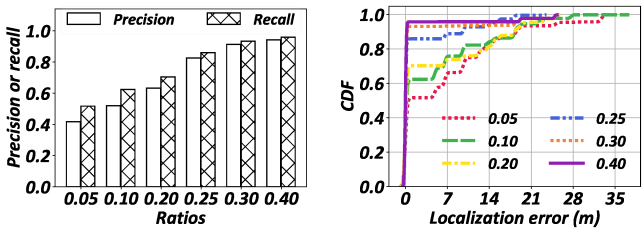
F. Ablation Study

This section evaluates the effectiveness of *OrchLoc*'s components in constructing a database for a new area. By default, the CGM is pre-trained in area A and fine-tuned using the training data from 30% of the locations in area B.



(a) CGM model (b) CSI and location representer

Fig. 21. Effectiveness of the proposed components in the CGM.



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

Fig. 22. Impact of the number of locations used in the turbo-training scheme.

1) Turbo-Training Scheme: We consider two alternatives to the turbo-training: (i) training CGM from scratch and (ii) pre-training CGM but fine-tuning only the location classifier.

Figure 21(a) illustrates that these two alternatives lead to location errors of 4.3 m and 8.9 m, respectively. The first alternative performs poorly because the CGM struggles to learn CSI-location relationships without pre-training, as different locations exhibit distinct spatial characteristics. The second alternative is less effective due to the limited modeling capacity of the location classifier [68], which consists solely of multiple FC layers. Its simplistic architecture and lack of physical location information hinder its ability to effectively generalize CSI-location relationships to new orchard areas.

2) CSI and Location Representers: To evaluate the performance of our CSI and location representers in area B, we experiment with six variations of LoDM inputs. "RA" utilizes the raw CSI vector $\mathbf{x}_0$ and the location ID's one-hot vector. "SE" omits the signal encoder, directly inputting the raw CSI vector. "LR" employs only the location ID's one-hot vector, excluding the location representer. "FE" provides the FFZ vector to LoDM while omitting the FFZ encoder. "FF" bypasses FFZ modeling, feeding the one-hot vector to the FFZ encoder before passing it to LoDM. Finally, "FU" incorporates both the CSI and location representers.

The quality of the generated data is quantified using the Fréchet Inception Distance (FID) [29], which measures the similarity between real and generated CSI fingerprints. Lower FID values indicate higher-quality generated CSI fingerprints.

Figure 21(b) illustrates that utilizing both the CSI and location representers significantly enhances performance. The average FID decreases from 49.7 to 10.9, accompanied by notable improvements in precision (from 67.4% to 94.1%) and recall (from 72.1% to 95.9%). Moreover, integrating FFZ modeling with the FFZ encoder proves more effective than employing either component separately. Using only the signal encoder results in an FID of 19.9 and a *precision* of 86.5%.

3) Location-Aware Diffusion Model: In LoDM, the attention layer is used to integrate location information into the denoise module. An alternative approach involves concatenating the location information with the intermediate vector of the U-Net network in the noise predictor, referred to as "LA" in Figure 21(b). Results indicate that employing the attention layer reduces the average FID, demonstrating a more effective integration of location information into LoDM. This enhances the model's ability to capture the CSI-location relationship, thereby improving CSI generation across different locations.

4) Synthetic Fingerprint Quality Across Rounds and Areas: In our previous ablation study, we confirm that our design components effectively learn the relationship between CSI fingerprints and locations. Given a location ID, *OrchLoc* generates high-quality corresponding CSI fingerprints. However, its capability for temporal and spatial adaptation across different rounds and areas requires further validation.

To this end, we analyze the impact of the turbo-training scheme on localization accuracy across multiple data collection rounds and areas. Specifically, as shown in Figures 15–18, we train the classifier using CGM-generated synthetic CSI data and evaluate it on real-world collected data across different locations and collection rounds. If the synthetic data does not align with the real-world collected data, the turbo-training scheme cannot improve accuracy [69]. Figures 15–18 indicate that our turbo-training scheme significantly enhances localization accuracy, demonstrating that CGM-generated fingerprints generalize well across temporal and spatial variations. This confirms that the generated synthetic data exhibits a distribution similar to real data, validating its high quality.

Temporal Adaptation: Figures 15 and 16 illustrate that CGM-generated fingerprints adapt well across multiple rounds. Without turbo-training, a classifier trained at round $r1$ suffers performance degradation in later rounds due to environmental changes. By incorporating CGM-synthesized data through turbo-training, the classifier preserves high accuracy in subsequent rounds. However, after significant seasonal transitions, as observed between $r7$ and $r8$, recalibration with real data becomes necessary, highlighting the need for periodic updates when substantial distribution shifts occur.

Spatial Adaptation: Figures 17 and 18 confirm the ability of CGM-generated fingerprints to generalize across different areas. In the first orchard, the classifier maintains high accuracy across multiple areas (B, C, D, E, F) with our turbo-training. However, in the second orchard, which has a slightly different tree layout, performance declines, even when turbo-training is applied with 30% of the locations' training data. This suggests that while CGM-synthesized fingerprints enhance generalization, significant structural variations across regions still necessitate additional recalibration.

G. Parameter Study

Unless specified otherwise, the turbo-training involves pre-training the CGM in area A and fine-tuning it with training data from 30% of the locations in area B. The accuracy of the trained classifier is evaluated on the testing set of area B.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3587206

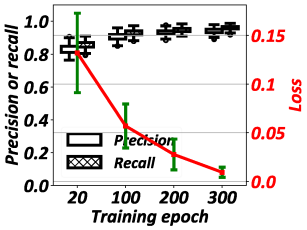


Fig. 23. Impact of training epochs on the noise predictor.

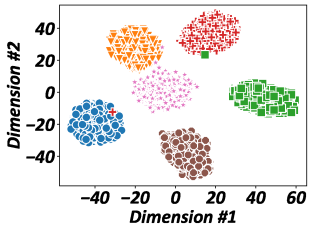


Fig. 24. Impact of the position offset of LoRa nodes around a tree.

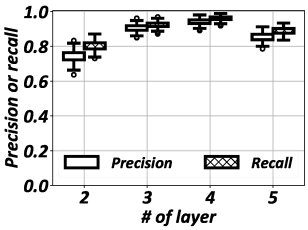


Fig. 25. Impact of the number of sampling layer in the noise predictor.

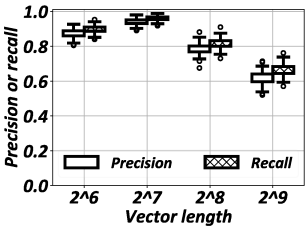


Fig. 26. Impact of the hidden length of the signal vector $\mathbf{z}_0$.

1) Ratios of Fine-Tuning Data: We explore the impact of varying fine-tuning location ratios from 5% to 40%. As illustrated in Figure 22, performance improves as the number of locations used for fine-tuning the CGM increases. *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% result in significantly lower performance, with only 51.9% precision at a 10% ratio. Thus, deploying sensor nodes at a minimum of 30% of the locations in a new area is essential to maintaining high performance.

OrchLoc requires the deployment of LoRa sensor nodes at 30% of the trees in an area. This relatively dense deployment aligns with current trends in precision agriculture, where fine-grained measurements of orchards are becoming increasingly essential for optimal farm management and sustainability [1], [2]. In the future, the challenge will be to maintain high accuracy while potentially reducing the number of required deployment locations. Several promising research directions include: (i) Incorporating information from other modalities, such as satellite images, to provide additional contextual data that could enhance the learning process; (ii) Utilizing active learning strategies to intelligently select the most informative locations; (iii) Investigating hybrid models that combine both supervised and unsupervised techniques, which may improve accuracy with fewer deployment locations.

In addition to the ratio of deployed nodes, the spatial distribution of these nodes also affects performance. Our experiments consider a scattered distribution of sensor nodes, which ensures coverage across the entire orchard. A scattered deployment aligns with principles of spatial sampling theory, which emphasize the importance of maximizing spatial coverage, minimizing sampling bias, and capturing local spatial variability for accurate representation of the propagation environment [70]. Such a deployment enables the CGM to learn from a diverse set of signal propagation paths, reflecting interactions with different media—air, foliage, and ground—and enhancing the generalizability of the generated fingerprints to areas without direct sensor coverage. In contrast, a concentrated distribution of sensors in a limited region risks overfitting to local propagation characteristics and reducing the system's ability to generalize to the rest of the orchard. Therefore, guided by spatial sampling theory, deploying sensor nodes in a scattered manner is more effective in providing the fine-tuning data needed for robust and accurate localization.

2) Training Epoch of Noise Predictor: To assess the impact of training epochs on performance, we pre-train and fine-tune the CGM with varying epochs, ranging from 50 to 300. As shown in Figure 23, increasing the number of training epochs improves both precision and recall. This improvement is attributed to the noise predictor becoming more effective at accurately estimating noise in the noisy signal vector $\mathbf{z}_t$, as evidenced by the decreasing validation loss in Figure 23.

3) Position of LoRa Node: Typically, the LoRa node is placed 2.0 m to the right of each tree during CSI data collection. To evaluate the impact of node position offset on CSI data distribution, the node is placed at five different locations around the predetermined spot for each tree, repeated for six trees. Figure 24 presents the CSI data visualized via t-SNE [44]. The data exhibit a clear clustering effect, where data from the same tree cluster together, while those from different trees remain distinct. This indicates that, despite position offsets at each location, the data collected near a given tree maintain a similar distribution.

4) Number of Sampling Layers in Noise Predictor: The noise predictor incorporates multiple sampling layers within its contracting and expansive paths. We vary the number of these layers from 2 to 5 in our experiments. Figure 25 shows that increasing the number of layers from 2 to 4 enhances precision from 74.2% to 94.1%. However, further increasing to 5 layers reduces precision to 70.5%. While fewer layers may fail to adequately extract latent CSI-location relationships, an excessive number of layers can significantly reduce the vector length, potentially leading to information loss.

5) Length of Signal Vector: To evaluate the impact of signal vector length on performance, we pre-train and fine-tune the CGM using vector lengths of 2^6, 2^7, 2^8, and 2^9. Figure 26 illustrates that recall improves from 89.7% at 2^6 to 95.9% at 2^7 but declines to 81.5% and 66.3% as the length increases to 2^8 and 2^9, respectively. A shorter vector length results in a coarse representation of the CSI vector, whereas a longer length significantly increases the model size, from 17.1 MB at 2^7 to 178.4 MB at 2^9. This larger model size requires extensive training data, which is challenging with limited data availability, ultimately leading to degraded performance.

H. Overhead of OrchLoc

Robot: For a single localization inference, the robot transmits eight two-byte packets and receives the localization result from the gateway, a process that takes 1.4 s at SF10 (the lowest data rate permitted in the US). Given the robot's velocity of 0.3 m/s [71], typical of agricultural robots, this corresponds to a movement of 0.42 m within the 1.4 s interval. As shown in Figure 24, such a minor positional shift has a negligible impact on CSI data quality. For robots operating at higher speeds, incorporating periodic 1.4 s pulses for localization helps

mitigate potential accuracy degradation. In terms of power consumption, at 0.4 W over 1.4 s, the energy consumption per localization inference is 0.56 J. Thus, *OrchLoc* is a low-power, sustainable solution well-suited for agricultural deployments.

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 consumption. While this increases the risk of packet collisions, it can be mitigated through careful scheduling, leveraging the week-long high accuracy of the location classifier, and employing MAC protocols or collision resolution techniques [39], [72], [73].

Gateway: The gateway is responsible for fine-tuning the CGM, generating CSI data, and performing inference. We evaluate memory requirements and execution time. As shown in Table II, the location classifier has minimal memory usage (0.02 MB) and a running time of (31.4 ms).

Labor: *OrchLoc* requires only a one-time measurement of orchard parameters for 3D modeling. This includes tree parameters (height, trunk height, and canopy width), orchard layout (row and column spacing), and the positions of sensors and gateways, which are easily determined due to the uniform deployment throughout the orchard.

TABLE II
MEMORY OCCUPANCY (UNIT: MB) AND TIME CONSUMPTION (UNIT: MS).

	CSI representer	LoDM	Classifier
Memory	0.15	17.1	0.02
Time	77.2 ± 14.8	843.3 ± 65.2	31.4 ± 9.1

IX. Related Work

A. Agricultural Robot Navigation

GPS/INS integrates INS data with GPS data using a Kalman filter-based algorithm for robot navigation [12], [74]. For example, Neurl-KF [12] employs a neural network to estimate a robot's velocity and location from raw inertial data, refining these estimates with GPS data through a Kalman filter to enhance navigation accuracy. These algorithms rely heavily on GPS precision to correct drift in INS sensors. However, in orchard environments, GPS accuracy is significantly degraded due to signal blockage from crop canopies, posing a major challenge [12], [74]. Although RTK GPS improves localization accuracy, its deployment requires substantial infrastructure, including base stations throughout the orchard, making it economically impractical for many agricultural operations. In contrast, our system leverages the existing LoRa network infrastructure in orchards to provide a reliable and cost-effective positional reference for INS sensors.

B. Localization by LoRa Network

The limited bandwidth of LoRa, at only 125 kHz [75], [76], poses challenges for directly applying conventional localization algorithms, which are typically designed for Wi-Fi, Bluetooth, or cellular networks [77] due to their reliance on higher bandwidths [78], [79]. 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 reducing accuracy. Moreover, data transmission and service fees make such approaches cost-prohibitive for agriculture.

Current research on localizing LoRa nodes employs techniques such as Time Difference of Arrival (TDoA) [23], [80], Angle of Arrival (AoA) [81], or path loss models [82]. For example, Seirios [81] achieves a median localization error of 4.4 m across a 6,000 m^2 area using AoA-based localization with multiple dual-antenna gateways. However, these methods typically require at least three gateways and depend on a direct signal path between the node and gateway, conditions that are often unavailable in orchard environments. In contrast, our system introduces a fingerprinting-based localization approach using a single gateway, overcoming the limitations of multi-gateway setups and reliance on direct signal paths.

Existing LoRa fingerprinting-based localization approaches typically use RSSI as a fingerprint [83], [84], [85], [86]. These methods match RSSI measurements from multiple gateways with a database to determine a LoRa node's location. However, they assume that a packet is received by multiple gateways, which is infeasible in orchards. In contrast, *OrchLoc* enables in-orchard fingerprinting-based localization by extracting CSI fingerprints using a single dual-antenna gateway.

C. CSI-based Fingerprinting in Wi-Fi

CSI-based fingerprinting is a key technique for localization with Wi-Fi [25], [26], [27]. These methods typically rely on either amplitude or phase independently, neglecting the combined information between them. *OrchLoc* incorporates a complex-valued classifier to effectively capture the latent relationships between the amplitude and phase of CSI data.

D. Crowdsourcing in Wi-Fi

Crowdsourcing has been applied to collect fingerprints for Wi-Fi-based indoor localization [87], [88]. For instance, Zee [87] leverages smartphones' inertial sensors to map fingerprints onto indoor layouts during routine user movement. EZ [88] integrates crowd-sourced Wi-Fi measurements with wireless propagation physics, employing a log-distance path loss model and a genetic algorithm to determine access point locations and propagation characteristics, enabling localization through trilateration. However, crowdsourcing is impractical in orchards, as it would require numerous robots to navigate the area and precisely track trajectories via INS sensors, both of which pose significant challenges in this environment.

E. Generative Models for Wireless Signals

Recent studies have explored generative models for wireless signal modeling [89], [90], [91]. For instance, NeRF^2 [89] segments 3D spaces into voxels for signal strength estimation, but this approach is impractical in large orchards due to the massive number of voxels required. WiNeRT [90] computes received signals based on environmental meshes, which are unavailable in orchards. In contrast, this paper adapts diffusion models to synthesize CSI data across different locations.

F. Transfer Learning-Based Localization

Several studies have explored transfer learning for CSI-based localization across multiple environments [55], [56], [57]. For example, Foliadis *et al.* [55] introduce a multi-environment meta-learning approach to adapt deep learning models across environments, improving spatial adaptation but not addressing temporal stability. Wang *et al.* [56] propose DJMDAN, which aligns feature distributions to mitigate domain shifts. Li *et al.* [57] explore CNN-based transfer learning for Wi-Fi localization, reusing feature extraction layers with minimal retraining. Jiao *et al.* [58] propose a few-shot meta-learning-based CSI fingerprinting method with task-weighted loss. Although these transfer learning-based methods focus on diverse environments, they do not address long-term localization stability. Essentially, *OrchLoc* is not a transfer learning model for different environments; instead, it operates within similar tree layout areas in an orchard while maintaining localization accuracy over time.

X. Discussion and Future Work

A. Why Diffusion Models

We choose diffusion models for their ability to iteratively refine noisy data into structured outputs, leveraging an autoregressive process similar to large language models (LLMs). This iterative approach progressively captures relationships between CSI fingerprints and location IDs. Existing diffusion model-based works have demonstrated their high-fidelity data generation capabilities [29]. Compared to alternative generative approaches such as Generative Adversarial Networks (GANs) [92] and Variational Autoencoders (VAEs) [68], diffusion models offer several advantages.

• First, they provide stability and training efficiency by avoiding the challenges of mode collapse and instability inherent in GANs. Since diffusion models do not rely on an adversarial setup, they follow a simpler denoising trajectory.

• Second, they offer superior expressiveness and data fidelity. Unlike VAEs, which often produce blurry outputs due to a trade-off between reconstruction quality and latent space regularization, diffusion models preserve fine-grained structural details of the original data.

• Third, their autoregressive nature simplifies data generation by decomposing it into multiple steps, unlike GANs and VAEs, which generate data in a single step.

• Finally, the denoising mechanism of diffusion models enhances robustness to random noise, making them more resilient to environmental variations than GANs and VAEs.

B. Connection with Transfer Learning-Based Localization

Existing transfer learning-based localization methods focus on spatial adaptation across multiple environments [55], [56], [57], [58]. *OrchLoc* differs in two key aspects.

• Unlike transfer learning-based methods that adapt models across environments, *OrchLoc* operates within a single orchard environment with a structured tree layout to maintain spatial homogeneity. Essentially, *OrchLoc* is not designed to function across environments. Future research could explore adapting *OrchLoc* to complex environments using transfer learning.

• Second, existing transfer learning-based approaches do not address long-term temporal variations in the fingerprint database, known as the aging problem. In contrast, our work integrates a generative diffusion model and a turbo-training scheme to stabilize fingerprints over time.

C. Why Not Complex Gaussian Noise in LoDM

We choose real-valued Gaussian noise instead of complex Gaussian noise in our LoDM framework to avoid fundamental challenges that could compromise model effectiveness. Real-valued Gaussian noise ensures a well-structured forward process, enabling direct reparameterization and stable noise estimation, whereas complex Gaussian noise complicates the reverse denoising process, potentially leading to instability. Additionally, complex Gaussian noise requires modeling both real and imaginary covariance terms, significantly increasing computational overhead and breaking key assumptions necessary for efficient sampling. Furthermore, our neural network already captures essential relationships between amplitude and phase in CSI data, eliminating the need for an explicit complex Gaussian formulation. While complex Gaussian noise could offer potential benefits, we plan to explore its theoretical and practical implications in future work.

XI. Conclusion

We present *OrchLoc*, a fingerprinting-based in-orchard localization system that achieves tree-level accuracy using a single LoRa gateway. *OrchLoc* leverages CSI-based fingerprints and incorporates a complex-valued FC block as the classifier for precise location estimation. To efficiently build and maintain the CSI fingerprint database across the orchard, we design a turbo-training scheme powered by the CGM, significantly reducing labor costs. Extensive experiments validate the effectiveness and robustness of *OrchLoc*.

This work does not raise any ethical concerns. Code is available at https://github.com/kangyang73/orchloc_mobisys24.

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.

References

[1] R. Gebbers and V. I. Adamchuk, "Precision agriculture and food security," *Science*, vol. 327, no. 5967, pp. 828–831, 2010.
[2] R. Bongiovanni and J. Lowenberg-DeBoer, "Precision agriculture and sustainability," *Precision agriculture*, vol. 5, pp. 359–387, 2004.
[3] K. Yang, "Advanced LoRa Networking and Sensing for IoT Applications," Ph.D. dissertation, University of California, Merced, 2024.
[4] K. Yang, Y. Chen, and W. Du, "OrchLoc: In-Orchard Localization via a Single LoRa Gateway and Generative Diffusion Model-based Fingerprinting," in *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, 2024, pp. 304–317.

[5] X. Li, W. S. Lee, M. Li, R. Ehsani, A. R. Mishra, C. Yang, and R. L. Mangan, "Spectral difference analysis and airborne imaging classification for citrus greening infected trees," *Computers and Electronics in Agriculture*, vol. 83, pp. 32–46, 2012.

[6] S. M. Bureau, A. J. Razungles, and R. L. Baumes, "The aroma of Muscat of Frontignan grapes: effect of the light environment of vine or bunch on volatiles and glycoconjugates," *Journal of the Science of Food and Agriculture*, vol. 80, no. 14, pp. 2012–2020, 2000.

[7] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-Level localization with a single WiFi access point," in *USENIX NSDI*, 2016.

[8] H. Chen, L. Zhang, D. Li, J. Xu, W. Yang, and Z. Yang, "REFLoc: A Resilient Evolutionary Fusion Framework for Robust Indoor Localization," *IEEE Transactions on Instrumentation and Measurement*, 2024.

[9] J. Yi, H. Wang, J. Zhang, D. Song, S. Jayasuriya, and J. 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*, vol. 25, no. 5, pp. 1087–1097, 2009.

[10] W. Luo, Q. Song, Z. Yan, R. Tan, and G. Lin, "Indoor Smartphone SLAM With Acoustic Echoes," *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, pp. 6634–6649, 2024.

[11] A. Abu and R. Diamant, "A SLAM Approach to Combine Optical and Sonar Information From an AUV," *IEEE Transactions on Mobile Computing*, vol. 23, no. 7, pp. 7714–7724, 2024.

[12] Y. Du, S. S. Saha, S. S. Sandha, A. Lovekin, J. Wu, S. Siddharth, M. Chowdhary, M. K. Jawed, and M. Srivastava, "Neural-Kalman GNSS/INS Navigation for Precision Agriculture," in *IEEE ICRA*, 2023.

[13] M. Brossard, A. Barrau, and S. Bonnabel, "RINS-W: Robust inertial navigation system on wheels," in *IEEE/RSJ IROS*, 2019.

[14] Y. Du, B. Mallajosyula, D. Sun, J. Chen, Z. Zhao, M. Rahman, M. Quadir, and M. K. Jawed, "A low-cost robot with autonomous recharge and navigation for weed control in fields with narrow row spacing," in *IEEE/RSJ IROS*, 2021.

[15] X. Cai, J. Xu, K. Deng, H. Lan, Y. Wu, X. Zhuge, and Z. Yang, "TrinitySLAM: On-board Real-time Event-image Fusion SLAM System for Drones," *ACM Transactions on Sensor Networks*, vol. 20, no. 6, pp. 1–22, 2024.

[16] D. Li, Y. Zhao, J. Xu, S. Zhang, L. Shangguan, Q. Ma, X. Ding, and Z. Yang, "Reshaping Edge-Assisted Visual SLAM by Embracing On-Chip Intelligence," *IEEE Transactions on Mobile Computing*, 2024.

[17] D. Li, Y. Zhao, J. Xu, S. Zhang, L. Shangguan, and Z. Yang, "edgeSLAM2: Rethinking Edge-Assisted Visual SLAM with On-Chip Intelligence," in *IEEE INFOCOM*. IEEE, 2024.

[18] D. Sacaleanu and I. Kiss, "Monitoring walnut orchards with LoRa technology," in *IEEE SIITME*, 2018.

[19] Y. Chen, K. Yang, Z. An, B. Holder, L. Paloutzian, K. M. Bali, and W. Du, "MARLP: Time-series Forecasting Control for Agricultural Managed Aquifer Recharge," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4862–4872.

[20] M. Preti, F. Verheggen, and S. Angeli, "Insect pest monitoring with camera-equipped traps: strengths and limitations," *Journal of pest science*, vol. 94, no. 2, pp. 203–217, 2021.

[21] H. Zhang, L. He, F. Di Gioia, D. Choi, A. Elia, and P. Heinemann, "LoRaWAN based internet of things (IoT) system for precision irrigation in plasticulture fresh-market tomato," *Smart Agricultural Technology*, vol. 2, p. 100053, 2022.

[22] W. Bakkali, M. Kieffer, M. Lalam, and T. Lestable, "Kalman filter-based localization for Internet of Things LoRaWAN end points," in *IEEE PIMRC*, 2017.

[23] D. Guo, C. Gu, L. Jiang, W. Luo, and R. Tan, "ILLOC: In-hall localization with standard lorawan uplink frames," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 1, pp. 1–26, 2022.

[24] N. Podevijn, D. Plets, J. Trogh, L. Martens, P. Suanet, K. Hendrikse, and W. Joseph, "TDoA-based outdoor positioning with tracking algorithm in a public LoRa network," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–9, 2018.

[25] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE transactions on vehicular technology*, vol. 66, no. 1, pp. 763–776, 2016.

[26] X. Wang, L. Gao, and S. Mao, "CSI phase fingerprinting for indoor localization with a deep learning approach," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1113–1123, 2016.

[27] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 316–327, 2018.

[28] X. Zhu, T. Qiu, W. Qu, X. Zhou, M. Atiquzzaman, and D. O. Wu, "BLS-Location: A Wireless Fingerprint Localization Algorithm Based on Broad Learning," *IEEE Transactions on Mobile Computing*, vol. 22, no. 1, pp. 115–128, 2021.

[29] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, 2020.

[30] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *ACM SIGGRAPH*, 2022.

[31] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996, vol. 2.

[32] K. Yang and W. Du, "LLDPC: A Low-Density Parity-Check Coding Scheme for LoRa Networks," *ACM Transactions on Sensor Networks*, vol. 20, no. 4, pp. 1–29, 2024.

[33] K. Yang, M. Liu, and W. Du, "RALoRa: Rateless-Enabled Link Adaptation for LoRa Networking," *IEEE/ACM Transactions on Networking*, 2024.

[34] K. Yang and W. Du, "LLDPC: A Low-Density Parity-Check Coding Scheme for LoRa Networks," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 193–206.

[35] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *PMLR ICML*, 2015.

[36] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[37] H. Qi and J. B. Moore, "Direct Kalman filtering approach for GPS/INS integration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 687–693, 2002.

[38] Y. Xie, J. Xiong, M. Li, and K. Jamieson, "mD-Track: Leveraging multi-dimensionality for passive indoor Wi-Fi tracking," in *ACM MobiCom*, 2019.

[39] X. Xia, N. Hou, Y. Zheng, and T. Gu, "Pcube: scaling lora concurrent transmissions with reception diversities," in *ACM MobiCom*, 2021.

[40] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity WiFi," in *ACM MobiCom*, 2015.

[41] J. Liu, W. Xu, S. Jha, and W. Hu, "Nephalai: towards LPWAN C-RAN with physical layer compression," in *ACM MobiCom*, 2020.

[42] D. Hafner, D. Tran, T. Lillicrap, A. Irpan, and J. Davidson, "Noise contrastive priors for functional uncertainty," in *UAI*, 2020.

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

[44] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[45] J. Li, I.-T. Lu, and J. S. Lu, "Hybrid fingerprinting and ray extension localization in NLOS regions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23 503–23 516, 2022.

[46] L. Zhang, N. Xiao, J. Li, and W. Yang, "Heterogeneous feature machine learning for performance-enhancing indoor localization," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, 2018, pp. 1–5.

[47] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, pp. 19–67, 2005.

[48] K. Yang, Y. Chen, X. Chen, and W. Du, "Link Quality Modeling for LoRa Networks in Orchards," in *ACM/IEEE IPSN*, 2023.

[49] L. Torresani and K.-c. Lee, "Large margin component analysis," in *NeurIPS*, 2006.

[50] V. Jakkula, "Tutorial on Support Vector Machine (SVM)," *School of EECS, Washington State University*, vol. 37, no. 2.5, p. 3, 2006.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[52] K. Yang, X. Zhao, J. Zou, and W. Du, "ATPP: A Mobile App Prediction System Based on Deep Marked Temporal Point Processes," *ACM Transactions on Sensor Networks*, vol. 19, no. 3, pp. 1–24, 2023.

[53] K. Yang, J. Zou, X. Zhao, and W. Du, "ATPP: A Mobile App Prediction System Based on Deep Marked Temporal Point Processes," in *IEEE DCOSS*, 2021.

[54] S. Demetri, G. P. Picco, and L. Bruzzone, "LaPS: LiDAR-assisted placement of wireless sensor networks in forests," *ACM Transactions on Sensor Networks*, vol. 15, no. 2, pp. 1–40, 2019.

[55] A. Foliadis, M. H. Castañeda, R. A. Stirling-Gallacher, and R. S. Thomä, "Transfer learning for csi-based positioning with multi-environment meta-learning," *arXiv preprint arXiv:2405.11816*, 2024.

[56] J. Wang, Y. Fu, H. Feng, and J. Wang, "Transfer learning for indoor localization algorithm based on deep domain adaptation," *Sensors*, vol. 23, no. 23, p. 9334, 2023.

[57] P. Li, H. Cui, A. Khan, U. Raza, R. Piechocki, A. Doufexi, and T. Farnham, "Deep Transfer Learning for WiFi Localization," in *IEEE Radar Conference (RadarConf)*, 2021.
[58] J. Jiao, X. Wang, C. Han, Y. Huang, and Y. Zhang, "Dynamic indoor fingerprinting localization based on few-shot meta-learning with csi images," *arXiv preprint arXiv:2401.05711*, 2024.
[59] Semtech, "Semtech SX1276 Datasheet," https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276, 2020, [Online].
[60] Arduino, "Arduino Uno Rev3," https://store-usa.arduino.cc/products/arduino-uno-rev3/?selectedStore=us, 2021, [Online].
[61] Nuand, "bladeRF 2.0 micro xA4," https://www.nuand.com/product/bladerf-xa4, year = 2023, note=[Online].
[62] R. Subbaraman, Y. Guntupalli, S. Jain, R. Kumar, K. Chintalapudi, and D. Bharadia, "BSMA: scalable LoRa networks using full duplex gateways," in *ACM MobiCom*, 2022.
[63] N. Hou, X. Xia, and Y. Zheng, "Don't miss weak packets: Boosting LoRa reception with antenna diversities," *ACM Transactions on Sensor Networks*, vol. 19, no. 2, pp. 1–25, 2023.
[64] United States Department of Agriculture National Agricultural Statistics Service, "2021 California Almond Acreage Report," https://www.almonds.com/sites/default/files/2022-04/2021_NASS_Acreage_Report.pdf.
[65] Dungmobile, "5941 IC GPS," http://dungmobile.vn/5941-ic-gps-iphone-xs-max-id3247.html, 2020, [Online].
[66] American Pistachio Growers., "Growing and harvesting American pistachios," https://americanpistachios.org/growing-and-harvesting.
[67] B. Zhang, H. Sifaou, and G. Y. Li, "CSI-fingerprinting indoor localization via attention-augmented residual convolutional neural network," *IEEE Transactions on Wireless Communications*, vol. 22, no. 8, pp. 5583–5597, 2023.
[68] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.
[69] Y. Yin, Z. Lin, M. Jin, G. Fanti, and V. Sekar, "Practical GAN-based synthetic IP header trace generation using NetShare," in *Proceedings of the ACM SIGCOMM 2022 Conference*, 2022.
[70] N. Cressie, *Statistics for Spatial Data*. John Wiley & Sons, 2015.
[71] I. Starostin, A. Eshchin, and S. Davydova, "Global trends in the development of agricultural robotics," *IOP Conference Series: Earth and Environmental Science*, vol. 1138, no. 1, p. 012042, 2023.
[72] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "LMAC: Efficient carrier-sense multiple access for LoRa," in *ACM MobiCom*, 2020.
[73] S. Tong, Z. Xu, and J. Wang, "CoLoRa: Enabling Multi-Packet Reception in LoRa Networks," *IEEE Transactions on Mobile Computing*, vol. 22, no. 6, pp. 3224–3240, 2021.
[74] N. Chebrolu, P. Lottes, T. Läbe, and C. Stachniss, "Robot localization based on aerial images for precision agriculture tasks in crop fields," in *IEEE ICRA*, 2019.
[75] N. Hou, X. Xia, Y. Wang, and Y. Zheng, "One shot for all: Quick and accurate data aggregation for LPWANs," *IEEE/ACM Transactions on Networking*, 2024.
[76] Y. Yang, Y. Chen, K. Yang, S. Yang, and W. Du, "Demo Abstract: Comprehensive Wireless Soil Component Sensing via VNIR and LoRa," in *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*, 2025, pp. 722–723.
[77] Y. Zhang, A. Y. Ding, J. Ott, M. Yuan, J. Zeng, K. Zhang, and W. Rao, "Transfer Learning-Based Outdoor Position Recovery With Cellular Data," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 2094–2110, 2020.
[78] J. Xiong, K. Sundaresan, and K. Jamieson, "Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *ACM MobiCom*, 2015.
[79] A. Eid, J. Zhu, L. Xu, J. G. Hester, and M. M. Tentzeris, "Holography-based target localization and health monitoring technique using UHF tags array," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14 719–14 730, 2021.
[80] 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].
[81] J. Liu, J. Gao, S. Jha, and W. Hu, "Seirios: leveraging multiple channels for LoRaWAN indoor and outdoor localization," in *ACM MobiCom*, 2021.
[82] Y. Lin, W. Dong, Y. Gao, and T. Gu, "Sateloc: A virtual fingerprinting approach to outdoor LoRa localization using satellite images," *ACM Transactions on Sensor Networks*, vol. 17, no. 4, pp. 1–28, 2021.
[83] J. Purohit, X. Wang, S. Mao, X. Sun, and C. Yang, "Fingerprinting-based indoor and outdoor localization with LoRa and deep learning," in *IEEE GLOBECOM*, 2020.
[84] R. Tian, H. Ye, and L. Sheng, "Indoor Localization Based on the LoRa Technology," in *MLICOM*, 2021.
[85] H. Zhu, K.-F. Tsang, Y. Liu, Y. Wei, H. Wang, C. K. Wu, and H. R. Chi, "Extreme RSS based indoor localization for LoRaWAN with boundary autocorrelation," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4458–4468, 2020.
[86] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, "Sigfox and LoRaWAN datasets for fingerprint localization in large urban and rural areas," *Data*, vol. 3, no. 2, p. 13, 2018.
[87] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *ACM MobiCom*, 2012.
[88] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *ACM MobiCom*, 2010.
[89] X. Zhao, Z. An, Q. Pan, and L. Yang, "NeRF^2: Neural Radio-Frequency Radiance Fields," in *ACM MobiCom*, 2023.
[90] T. Orekondy, P. Kumar, S. Kadambi, H. Ye, J. Soriaga, and A. Behboodi, "WiNeRT: Towards Neural Ray Tracing for Wireless Channel Modelling and Differentiable Simulations," in *ICLR*, 2023.
[91] G. Chi, Z. Yang, C. Wu, J. Xu, Y. Gao, Y. Liu, and T. X. Han, "RF-Diffusion: Radio Signal Generation via Time-Frequency Diffusion," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 77–92.
[92] D. Saxena and J. Cao, "Generative adversarial networks (GANs) challenges, solutions, and future directions," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–42, 2021.

Kang Yang received his B.E. degree in automation engineering from the School of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an, China, in 2016, and the M.E. degree in control engineering from School of Electronic and Information, Xi'an Jiaotong University, Xi'an, China, in 2019. He is currently pursuing a Ph.D. degree at the University of California, Merced. His research interests include mobile computing and wireless networking systems.

Yuning Chen received his B.E. degree in software engineering from the School of Software, Tsinghua University, Beijing, China, in 2021. He is currently pursuing a Ph.D. degree at the University of California, Merced. His research interests include networked systems and wireless sensor networks.

Wan Du is an Assistant Professor at the University of California, Merced. He was a Research Fellow at Nanyang Technological University, Singapore 2012-2017. He received the B.E. and M.S. degrees in Electrical Engineering from Beihang University, China, in 2005 and 2008, respectively, and a Ph.D. degree in Electronics from the University of Lyon (Ecole Centrale de Lyon), France, in 2011. His research interests include Internet of Things, distributed networking systems, and mobile computing.