Fingerprinting-based Indoor Localization with Deep Neural Networks

by

Xiangyu Wang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 10, 2022

Keywords: Indoor localization, Fingerprinting, WiFi, RFID, Deep learning, Internet of Things

Approved by

Shiwen Mao, Professor of Electrical and Computer Engineering
Thaddeus Roppel, Associate Professor of Electrical and Computer Engineering
Xiaowen Gong, Assistant Professor of Electrical and Computer Engineering
Mark Nelms, Professor and Chair of Electrical and Computer Engineering

Abstract

In recent years, more capabilities and applications have been added to existing wireless communication systems due to the rapid development of the Internet of Things(IoT) [1–8]. WiFi and RFID exhibits tremendous potential in this industry due to their prevalence and low-cost. Among the applications, indoor localization has been a popular field of research over the years, since it plays a vital role in resolving position-related challenges such as gesture recognition and human pose estimation. In the meantime, with the advancement of deep learning, researchers are attempting to integrate deep networks into indoor localization systems to take advantage of their superior ability to solve classification and regression problems. On the other hand, the fingerprint method emerges with its convenience and effectiveness, which transfers the localization problem into a feature matching to estimate the location of the signal. Thus, deep learning technique is a great complement to fingerprinting-based indoor localization systems. However, numerous intrinsic difficulties of fingerprinting-based localization systems remain unresolved even though the performance of indoor localization systems keeps improving with the iteration of deep networks. First, the distance between the stored fingerprints determines the minimum error of the fingerprinting-based localization system. To guarantee the lower-bound of the localization accuracy, as many fingerprints as possible have to be collected, which is laborious and time-consuming. Second, fingerprints are discrete signal space samples. As a result of the elimination of ambiguity between fingerprints, deep neural networks may produce counterintuitive location estimations, contrary to our expectations. To address such issues, the Deep Gaussian Process(DGP) is leveraged in this dissertation to generate a detailed radio signal map using a limited number of fingerprints. Then, the uncertainty information from DGP is adopted to train an LSTM model for enhancing the localization estimation by using the signal sequence. Furthermore, a novel network input, the hologram tensor, is employed for reserving the ambiguity between the fingerprints. In the last section, the threat of the adversarial attack to the fingerprints is investigated to promote the robustness of the localization system.

Acknowledgments

This work is supported in part by the US National Science Foundation under grants CNS-1702957, ECCS-1923163, CNS-1822055, ECCS-1923717, CNS-2107190, Auburn University RFID Lab, and by the Wireless Engineering Research and Education Center (WEREC) at Auburn University, Auburn, AL, USA.

I would like to express my deep gratitude to my advisor, Dr. Shiwen Mao, for providing me invaluable guidance in research. Besides my advisor, I am extremely grateful to my committee members, Dr. Xiaowen Gong, Dr. Thaddus Roppel, and Dr. Mark Nelms, for their insightful comments and support. My sincere appreciation also goes to Dr. Xuyu Wang, Dr. Jian Zhang, Dr. Yibo Lyu and Justin Patton, for their advice and inspiration. Last but not least, I would like to thank my parent and my girlfriend Anni Zhang, for their understanding, love, and continuing support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

The growing use of Internet of Things (IoT) devices has heightened interest in indoor location-based services. Relying on the progress of radio frequency communication systems [10–14], an increasing number of emerging indoor localization systems adopt radio frequency(RF) signals, such as WiFi, RFID, and Bluetooth, as the observation for indoor localization. Among the localization techniques, the fingerprinting method exhibits great performance with its convenience and effectiveness. The localization problem is recast as a feature matching problem. The unknown location may be deduced intuitively once the signal feature matched with a feature stored in the fingerprint database. Thus, deep learning techniques join the field of indoor localization with its outstanding performance in feature extraction and classification. However, the several intrinsic issues related to the fingerprinting method are not alleviated by the introduction of deep learning. First, high-accuracy fingerprinting-based localization relies on the density of fingerprints. A wardriving would be essential to the fingerprint collection, which is time-consuming and laborious. Second, the fingerprints, particularly those formed with angle of arrival, lack interpretability. The ambiguity between fingerprints could not be reserved during the generation of fingerprints. As a result, the transferability of the fingerprinting-based localization system is hampered. Furthermore, to increase localization accuracy, fingerprinting algorithms often focus on two angles: fingerprint and matching. Hence, position estimation wastes peripheral information such as indoor layout and trajectory history. Finally, because fingerprints are discrete signal samples from the signal space, most deep learning localization models ignore signal uncertainty, which should have been beneficial in resolving fingerprint updating concerns. In my dissertation, I would focus on these four aspects in order to improve the current fingerprinting-based indoor localization systems. Because the fingerprint method is strongly coupled with deep learning techniques in this dissertation, the threat of adversarial assaults on the deep learning-based localization system is also investigated.

1.2 Summary of Contributions

1.2.1 DeepMap: Indoor radio map construction and localization with deep Gaussian processes

DeepMap is a fingerprinting-based indoor localization system. Received signal strengths are leveraged as fingerprints to estimate the indoor location. To decrease the dependence to dense fingerprints and improve the distance resolution, a Deep Gaussian Process(DGP) is utilized to generate the radio signal maps. The system is prototyped with the COTS WiFi devices and evaluated in Auburn University Broun Hall. The experimental results demonstrate that a detailed radio signal map can be established with a limited number of fingerprints successfully.

1.2.2 MapLoc: LSTM-based Indoor Location Estimation using Confidence Interval Maps

As a follow-up project of DeepMap,the DGP model in the MapLoc recovered the received signal strengths of WiFi and the magnetic field strength and the corresponding uncertainty information. The collected fingerprints are augmented by sampling the distribution described with the uncertainties. Thus, the signal reliability is learnt by the deep networks with fluctuating signal measurements. To boost the localization accuracy, the trajectory information is considered in the MapLoc. We use the physical constraints of the indoor environment and the motion model to build the trajectory dataset. A LSTM model is trained to replace the traditional matching algorithm. With extensive experiments, a centimeter-level localization has been demonstrated in MapLoc.

1.2.3 MulTLoc: A Framework for Multiple RFID Tag localization Using RF Hologram Tensors with Deep Neural Networks

All previous systems rely on received signal strengths as measurements for indoor localization, whereas MulTLoc utilizes phase information collected from RFID systems to complete centimeter-level localization. Compared with received signal strengths, phase information is sensitive to the distance change between the receiver and transmitter. Thus, the received phase values are usually noisy, including the offsets resulting from the multipath and phase wrapping effects. To eliminate the offset and optimize the transferability of the localization system, the RF hologram tensors are innovatively used as the fingerprints to estimate the target location. Two representative deep networks are deployed in the proposed system to sanitize the noisy hologram tensor for location prediction. In MulTLoc, we treat the location estimation as a regression task, while the traditional fingerprint-based systems would solve the localization estimation as a classification task. Thus, the interpretability of the system is enhanced significantly.

1.2.4 AdvLoc: Adversarial Deep Learning for Indoor Localization

In the previous projects, the localization performance took advantage of improving with the progress of the deep networks. However, some counter-intuitive properties of deep networks have been exposed recently, which make networks vulnerable to adversarial attacks. In AdvLoc, the effects of six types of common adversarial attacks are analyzed in both white-box attack and black-box attack scenarios. The extensive experimental study exposed the threat of adversarial attacks to indoor localization systems. Furthermore, adversarial training is leveraged in AdvLoc to defend against first-order adversarial attacks and promote the robustness of localization systems.

Chapter 2

DeepMap: Indoor radio map construction and localization with deep Gaussian processes

2.1 Introduction

Location-based service has collect significant attraction [15–19] due to the popularity of mobile devices and wireless networks. However, the accurate location estimation for mobile devices using radio frequency(RF) signals is still a challenging problem because the radio signal propagate in indoor environments unpredictably (e.g. the multipath degrades the localization precision of lots of indoor localization systems [20–23]). To address the accuracy degradation resulted from the complex signal propagation, the fingerprinting-based localization approach has been one of the hot topics. The basic idea about the fingerprinting-based localization approach consists of an offline stage and an online stage. Fingerprints are collected and stored in the offline stage. They consist of exhaustive records for the surveillance area. In the online stage, the location estimation is obtained by comparing the newly collected records to the predefined fingerprints [24].

Owing to the low hardware requirement and the ubiquitousness, the received signal strength (RSS) of WiFi signals is leveraged as fingerprints in many proposed localization systems. It is in Radar that the RSS is utilized as fingerprints for the first time [25]. Moreover, Horus [26] leverages a probabilistic method to enhance the localization accuracy of a RSS based fingerprinting system. After that, the channel sate information (CSI) attracts attention from researchers because it includes fine-grained information estimated from each subcarrier [27–31]. However, the density of fingerprints is still the key factor that affects the accuracy of indoor localization significantly. To achieve high-accuracy localization, a wardrive is essential to the fingerprint collection, which is time-consuming and laborious.

To get rid of the dependency on the war-driving, the radio map is constructed with discrete training data in some proposed localization systems. The Gaussian process is a popular method to build the radio map.In the area of cellular networks, it is in GPSS that Gaussian process is used for generating radio maps for the first time. In GPSS, the distribution of signal strengths is modeled by the Gaussian process and the unknown location is estimated by maximizing a joint likelihood [32]. Furthermore, Gaussian process regression is also utilized for modelling the log-signal strengths in many types of wireless systems [33–38]. With the model regressed by Gaussian process, the distance between the mobile devices to APs would be inferred conveniently. Then the accurate location of mobile devices is obtained by triangulation However, to locate the mobile devices, the accurate locations of access points (APs) would also be necessary. In many practical scenarios, it is impossible to acquire the precise coordination of APs.

The radio map construction problem in RSS fingerprinting based localization methods is addressed in this chapter. First of all, the Gaussian process for radio map construction is investigated. The Gaussian process is capable of measuring the uncertainty in input RSS data over a continuous space, and it is depicted with the mean and covariance function. Also, it is a Bayesian nonparametric model. For the radio map construction problem, Gaussian process could be leveraged to regress the relationship between RSS measurement values and their corresponding locations. Furthermore, the Gaussian process shows an agreeable ability in representing data when training data is adequate. However, such ability of Gaussian process degrades dramatically when RSS radio maps are generated with reduced training data. In fact, Gaussian process is not effective for handling the non-stationary components of RSS values, because of the lack of fusion of kernels in Gaussian process for complex input data [39]. Thus, the Gaussian process leads to an unacceptable localization accuracy when it is trained with reduced training data.

In this chapter, we propose DeepMap to solve this problem, which is a **Deep** Gaussian process for indoor radio **Map** construction and location estimation. It is noteworthy that the method is not restricted to WiFi RSS values. The proposed method could also be applied in the systems leveraging other wireless signals, such RFID and BLE. Like traditional fingerprinting based localization methods, the DeepMap system includes an offline training stage and online localization stage. In the offline stage, the RSS values labeled by corresponding coordinates are passed into a two-layer deep Gaussian process model for modeling the relationship between RSS values and coordinates in a continuous space. Also, we develop an offline Bayesian training method for maximizing the marginal distribution of the observed RSS values to compute optimal hyperparameters, where a variational lower bound makes the problem tractable. Unlike Gaussian process, deep Gaussian process is capable of constructing a precise radio map with inadequate training data. The structural advantage of deep Gaussian process enhances the learning capacity of training complicated datasets associated with abstract information [40]. Therefore, the distribution of a small dataset for radio map construction could be better described with deep Gaussian process. In the online stage, a Bayesian method is leveraged to enhance localization precision. With radio maps generated by deep Gaussian process and newly measured RSS values from all available APs, the location estimation is obtained with maximum a posteriori (MAP) estimation.

The main contributions of this chapter includes:

- We propose DeepMap system, which first utilizes deep Gaussian process for radio map construction and indoor localization. Deep Gaussian process effectively overcomes the drawbacks of Gaussian process, which could not regenerate radio maps in detail with limited numbers of training data.

- A two-layer deep Gaussian process model is designed to regress the relationship between the RSS space and the location space; a Bayesian training method is deployed for optimize model parameters; and a Bayesian fusion method is utilized to boost localization performance.

- We validate the proposed DeepMap system in two indoor environments. Even though the Gaussian process is comparable to deep Gaussian process when 100% training data are leveraged to train schemes, DeepMap outperforms the Gaussian process when moderate training data is available.

In the remainder if the paper, the preliminaries and motivation is presented in Section 2.2. The DeepMap system design is introduced in Section 2.3 and the performance evaluation is covered in Section 2.4. Section 2.6 concludes this chapter.

2.2 Preliminaries and Motivation

As a kernel based Bayesian model, Gaussian process has been leveraged in regression and classification successfully [39]. With the help of Gaussian process, the uncertainty in input data distribution over a continuous space could be measured. Generally, a Gaussian process could be delineated by its covariance and mean function, which is a generalization of a multivariate Gaussian distribution.

For radio map construction problems, we could treat measured RSS values and corresponding locations as a a Gaussian process regression model, that is

$$s = f(x) + \epsilon, \tag{2.1}$$

where s is the measured RSS at location x, $f(x)$ represents the pure RSS at location x, and ϵ is the observation noise, which follows an i.i.d. (independent, identically distributed) Gaussian distribution with zero mean and variance σ_n^2. The Gaussian process model assumes that the RSS measurements s_p and s_q at two different positions x_p and x_q follow a joint Gaussian distribution with covariance $k(x_p, x_q)$, which is a kernel function for the two locations given by

$$k(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\left|x_p - x_q\right|^2\right), \tag{2.2}$$

where σ_f and l are the hyper-parameters, σ_f^2 represents the variance and l is a length scale, both of which describe the smoothness of the kernel function. The predicted RSS for an unknown position x_* can be obtained by

$$\Pr(f(x_*)|X, Z, x_*) = N(f(x_*); u_*, \sigma_*^2) \tag{2.3}$$

$$u_* = k_*^T(K + \sigma_n^2 I)^{-1} Z \tag{2.4}$$

$$\sigma_*^2 = k(x_*, x_*) - k_*^T(K + \sigma_n^2 I)^{-1} k_*, \tag{2.5}$$

where k_* is an $n \times 1$ vector of covariances between training locations X and x_*, K is the covariance matrix of training locations X, Z is the training observation values matrix. In addition, the hyper-parameters σ_f and l can be estimated by a maximum likelihood approximation.

The RSS radio map in Fig. 2.1 is constructed with Gaussian process. As we can see, all training data from Broun Hall dataset (see Section 2.4.1) are utilized to train the Gaussian process. Obviously, the bell-shaped RSS radio map is consistent with most of the ground truth RSS values. Thus, it verifies that Gaussian process could model the distribution of RSS values in an indoor environment and regress the relationship with adequate training data. However, the ability of Gaussian process in depicting RSS data distribution downgrades remarkably with inadequate training data. Fig. 2.2 shows a RSS radio map constructed with 20% training data by Gaussian process. We find that the RSS radio map in Fig. 2.2 tends to be a plain. Clearly, most of rises and falls in Fig. 2.1 is lost in Fig. 2.2, even though the upper-right corner is still the highest area in Fig. 2.2. In other words, the non-stationary components of RSS values are lost in the radio map constructed by Gaussian process, because of the lack of fusion of kernels in Gaussian process for complex input data. Therefore, this coarse RSS radio map resulted from the deficiency of Gaussian process hampers the localization accuracy in the online stage. To alleviate the problem, a DeepMap system is proposed for RSS radio map construction using *deep Gaussian process* in the next section.

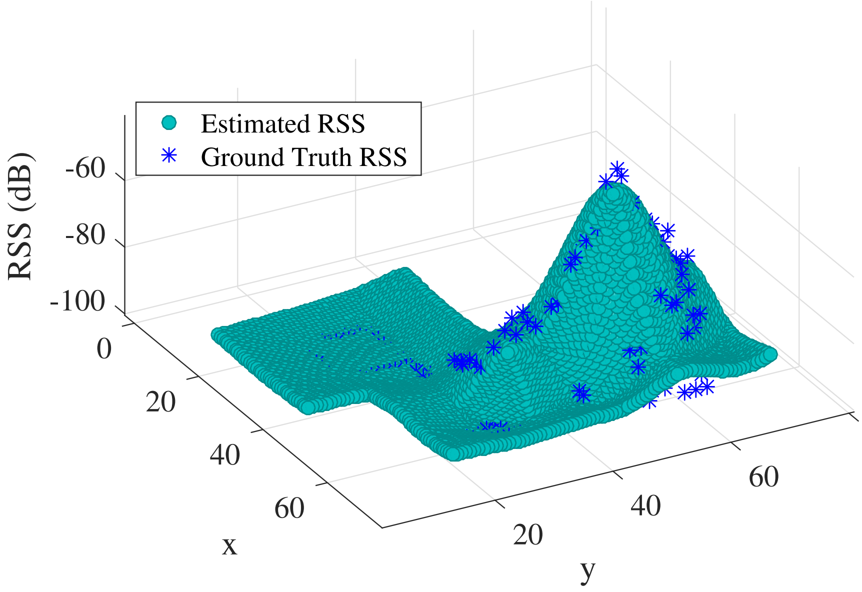


Figure 2.1: The constructed RSS radio map using 100% training data with Gaussian process.

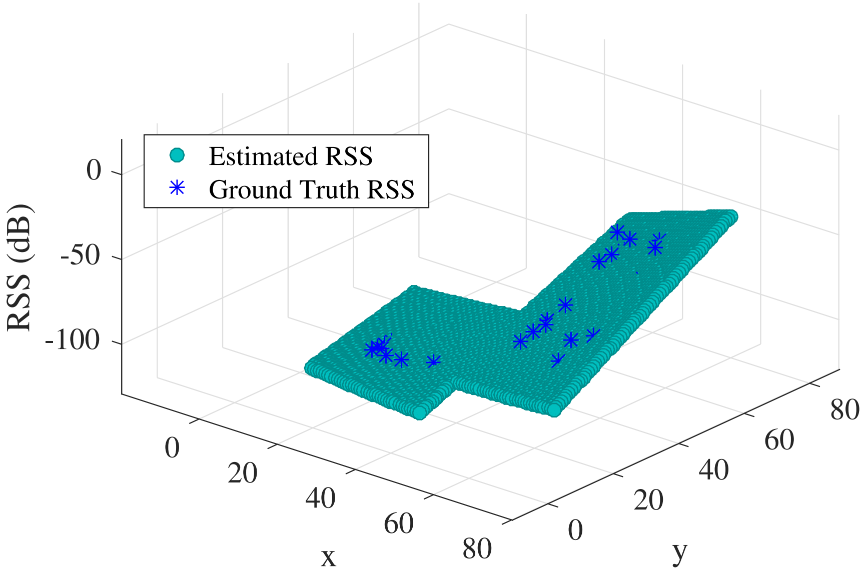


Figure 2.2: The constructed RSS radio map using 20% training data with Gaussian process.

2.3 The DeepMap System

2.3.1 DeepMap System Architecture

Fig. 2.3 shows the architecture of the DeepMap system. The DeepMap system is a fingerprinting based indoor localization method, which consists of two stages: the offline training stage

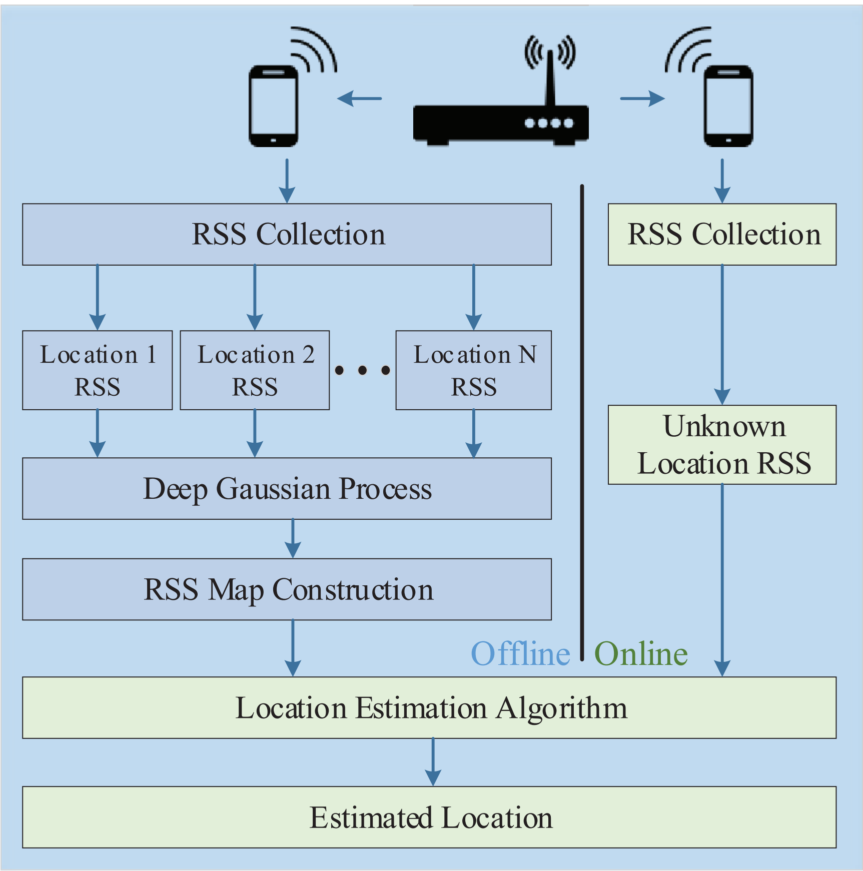


Figure 2.3: The DeepMap system architecture.

and the online localization stage. In the offline stage, we labeled the RSS values from training positions with corresponding coordinates. In each training location, RSS values are collected from as many as possible available APs to enhance localization accuracy. To guarantee the RSS records from all training locations are in the same size, we collect all potential RSS readings. For some specific locations, the corresponding RSS readings are set to a -99 dBm when the RSS values are unavailable. Therefore, a training dataset is generated with all the RSS records and the corresponding location labels. To construct RSS radio maps of the indoor environment, we employ a deep Gaussian process for regressing the training dataset. The well-trained model(reconstructed map) is saved in a database for future use, which describes the relationship between RSS values and location labels in a continuous space.

In the online stage, we collect RSS values from unknown locations and contrast them with the RSS values in the constructed radio maps. Then, the similarities between the measured RSS values and the RSS values in the radio maps are calculated. By synthesizing the similarities obtained from the radio maps of APs, the location for corresponding RSS values could be inferred using a Bayesian fusion method. Unlike the traditional fingerprinting methods which save original RSS readings as fingerprints or the autoencoder based methods that leverage a bunch of well-trained weights as fingerprints [27–29,41], the deepMap system has two different storage strategies. Depending on the specification of user devices, users could store the well-trained model to reconstruct radio maps in the online stage if the disk space is a limited resource for user devices. Alternatively, the constructed radio maps could be saved in the disk directly to accelerate the localization process. Furthermore, the resolution of the constructed radio maps could be variable. In fact, a high-resolution map offers a higher localization precision at the cost of localization speed, while a low-resolution map contributes to a coarse but fast localization. In following sections, we will show that the agreeable localization results would be obtained by the proposed DeepMap system even with a low-resolution map, compared to the Gaussian process method.

2.3.2 Deep Gaussian Process for Radio Map Construction

We propose a deep Gaussian process for radio map construction with RSS values, which can be represented by a graphical model with three different sets of nodes, including the leaf nodes, the intermediate latent nodes, and the parent nodes [40]. For radio map construction, the leaf nodes represent RSS values $Y \in \Re^{N \times D}$, where N and D are the number of training locations and the number of APs, respectively. The intermediate latent nodes are defined as $H \in \Re^{N \times Q}$, where Q is the size of the intermediate latent nodes in this layer. These latent nodes cannot be observed in the training phase. For the DeepMap system, we consider one intermediate latent layer to have a deep Guassian process. Let $X \in \Re^{N \times M}$ denote the parent nodes, where M is the input size. Parent nodes X represent the training locations.

It can be shown that the proposed deep Gaussian process for radio map construction is a generative model for regression. This generative process can be formulated by

$$h_{nq} = f_q^H(x_n) + \epsilon_{nq}^H,\ q = 1, 2, ..., Q,\ x_n \in \Re^M \tag{2.6}$$

$$y_{nd} = f_d^Y(h_n) + \epsilon_{nd}^Y,\ d = 1, 2, ..., D,\ h_n \in \Re^Q, \tag{2.7}$$

where $f^H \sim GP(\mathbf{0}, k^H(X,X))$ and $f^Y \sim GP(\mathbf{0}, k^Y(H,H))$ are Gaussian processes, and the intermediate nodes H connect the two Gaussian processes. Note that these two Gaussian processes only depend on the covariance function k for different inputs, where is chosen to be the automatic relevance determination (ARD) covariance function k, that is

$$k(x_i, x_j) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q(x_{i,q} - x_{j,q})^2\right), \tag{2.8}$$

where σ is the hyperparameter and w_q is the weight for latent node q. Irrelevant dimensions can be removed by setting their weights to zero.

2.3.3 Offline Bayesian Training

The objective of Bayesian training is to maximize the marginal distribution of observed RSS values Y to determine optimal hyperparameters, which is formulated as

$$\max\ \log p(Y) = \log \int_{X,H} p(Y|H)p(H|X)p(X), \tag{2.9}$$

Because of the nonlinear functions for H and Z, it is not easy to solve the integral in (2.9) with the maximum likelihood method. In DeepMap, we apply Jensen's inequality to achieve a variational lower bound for the above marginal distribution $L \leq \log p(Y)$, given by

$$L = \int_{F^Y,H,F^H,X} Q \log\left(\frac{p(Y, F^Y, H, F^H, X)}{Q}\right), \tag{2.10}$$

where Q is the variational distribution, and the term $p(Y, F^Y, H, F^H, X)$ is given by

$$\begin{aligned} &p(Y, F^Y, H, F^H, X) \\ &= p(Y|F^Y)p(F^Y|H)p(H|F^H)p(F^H|X)p(X). \end{aligned} \tag{2.11}$$

In fact, the above integral is still intractable due to the nonlinearity in both $p(F^Y|H)$ and $p(F^H|X)$. Consider the probability space with K auxiliary pseudo-inputs $\bar{H} \in \Re^{K\times Q}$ and $\bar{X} \in \Re^{K\times M}$ [42], whose function values are $U^Y \in \Re^{K\times D}$ and $U^H \in \Re^{K\times Q}$, respectively.

Then, we can derive the augmented probability space, as

$$\begin{aligned}&p(Y, F^Y, H, F^H, X, U^Y, U^H, \bar{H}, \bar{X})\\ =\ &p(Y|F^Y)p(F^Y|U^Y, H)p(U^Y|\bar{H})\\ &\cdot p(H|F^H)p(F^H|U^H, X)p(U^H|\bar{X})p(X).\end{aligned} \tag{2.12}$$

To remove the nonlinear items $p(F^Y|U^Y, H)$ and $p(F^H|U^H, X)$, the variational distribution Q is defined as

$$\begin{aligned}Q = &\ p(F^Y|U^Y, H)q(U^Y|\bar{H})q(H)\\ &\cdot p(F^H|U^H, X)q(U^H|\bar{X})q(X),\end{aligned} \tag{2.13}$$

where $q(U^Y|\bar{H})$ and $q(U^H|\bar{X})$ are free-form variational distributions, and $q(H)$ and $q(X)$ are Gaussian.

According to (2.11) and (2.13), we can update the variational lower bound for (2.10), as

$$L = \int Q \log\left(\frac{p(Y|F^Y)p(U^Y|\bar{H})p(H|F^H)p(U^H|\bar{X})p(X)}{q(U^Y|\bar{H})q(H)q(U^H|\bar{X})q(X)}\right),$$

where the integration is with respect to $\{F^Y, H, F^H, X, U^H, U^Y\}$. By grouping the variables for Y and H, we can rewrite the variational lower bound as

$$L = s_Y + s_H - q(H)\log(q(H)) - \mathrm{KL}(q(X)||p(X)), \tag{2.14}$$

where KL is the Kullback-Leibler divergence, s_Y is given by

$$s_Y = \mathbb{E}_{p(F^Y|U^Y,H)q(U^Y|\bar{H})q(H)}\left(\log p(Y|F^Y) + \log\frac{p(U^Y|\bar{H})}{q(U^Y|\bar{H})}\right),$$

and s_H is given by

$$s_H = \mathbb{E}_{p(F^H|U^H,X)q(U^H|\bar{X})q(X)}\left(\log p(H|F^H) + \log\frac{p(U^H|\bar{X})}{q(U^H|\bar{X})}\right).$$

We can see that both s_Y and s_X are Gaussian densities, which are thus tractable. In fact, Bayesian training for deep Gaussian process can optimize the above variational lower bound to seek the optimal hyerparameters for the deep Gaussian process, the inducing points ($\bar{H}$ and $\bar{X}$), and the variational parameters [40].

The constructed RSS radio map shown in Fig. 2.4 is generated by deep Gaussian process with 100% training data in the same Broun Hall dataset. Even though the similar bell-shape surface is also created by Gaussian process, deep Gaussian process brings more details on the bell-shape surface. For example, a slight fluctuation, which is closed to the coordinate origin, is delineated by deep Gaussian process in Fig. 2.4, however the corresponding area in Fig. 2.1 tends to be a plain surface, which is constructed by Gaussian process. In the Fig. 2.5, the RSS radio map is constructed by deep Gaussian process with 20% training data. Clearly, the bell-shape outline is also retained from the radio map generated by 100% training data, even though only 20% training data is utilized. Additionally, the surface contains most of the non-linear characteristics. It is safe to say that deep Gaussian process can handle non-stationary components comparing with the plain-like surface constructed by Gaussian process in Fig. 2.1. Moreover, nonlinear characteristics are also reproduced with only a few training data, because deep Gaussian process has a deep and heterogeneous nonlinear structure, which is more effective for complex training data. Thus, the radio map constructed by deep Gaussian process captures more detailed information of the real RSS values distribution for indoor environments, which contributes to improving localization precision considerably.

2.3.4 Online Phase

In the online localization stage, a Bayesian method is leveraged to estimate the location of a mobile device the newly measured RSS values from D APs and the constructed radio maps. We grid the RSS radio map to obtain T reference positions. The size of T is decided by the resolution of the RSS radio map. The pseudocode for the online localization estimation is presented in algorithm 1. The input to the algorithm 1 are the newly measured RSS values v_j, the constructed radio map r^j, the number of APs D and the number of reference points M. In the DeepMap system, we assume that the likelihood function $p(v_j|l_i)$ is a Gaussian function.

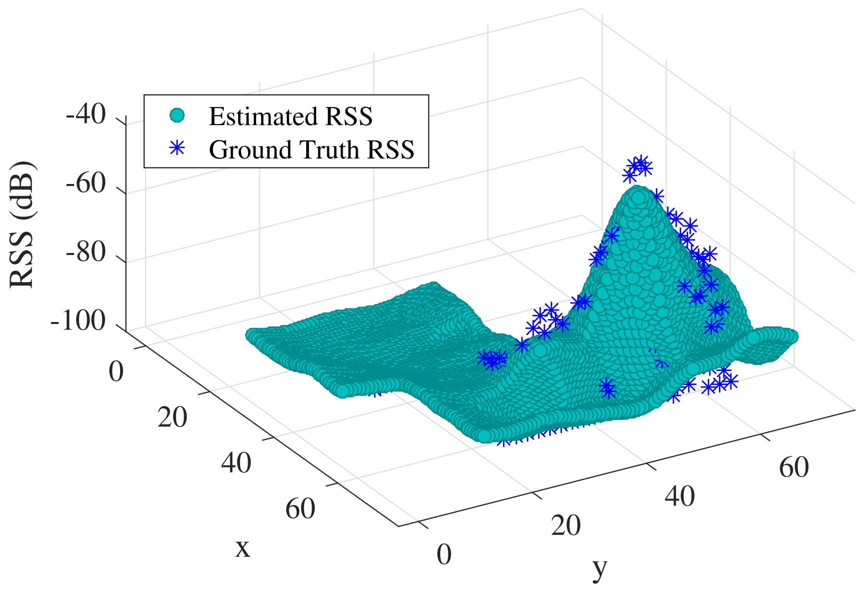


Figure 2.4: The constructed RSS radio map using 100% training data with deep Gaussian process.

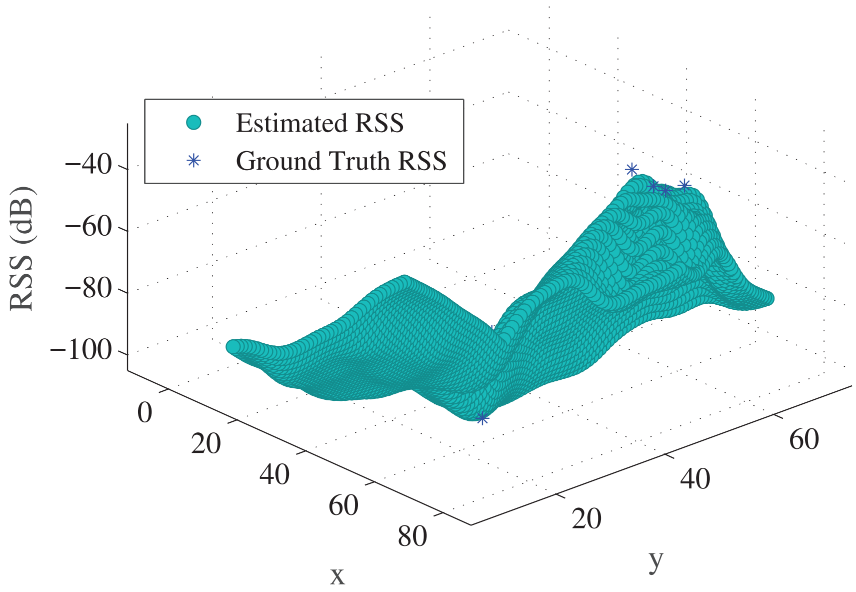


Figure 2.5: The constructed RSS radio map using 20% training data with deep Gaussian process.

Thus, the similarity between the measured RSS value v_j and the discrete data $r_{l_i}^j$ at location l_i in the radio map from AP j is computed in step 6 [27]. Here, σ^2 is the variance and λ is the parameter of the variance of the input RSS values. Based on the likelihood function, the

Algorithm 1 Pseudocode for Online Localization

```
Input: the measured RSS values $v_j$, the constructed radio map $r^j$, the number of APs $D$, and the number of reference points $M$;
Output: the estimated location $\hat{l}$;
//$j$ denotes the index of AP
//$i$ represents the index of the reference points in the radio map $r^j$
for $j = 1 : D$ do
    for $i = 1 : T$ do
        //compute the likelihood function $p(v_j|l_i)$
        $p(v_j|l_i) = \exp\left(-\frac{1}{\lambda\sigma^2}\left\|v_j - r^j_{l_i}\right\|\right)$
    end for
    //compute the posterior probability $p(l_i|v_j)$
    $p(l_i|v_j) = \frac{p(v_j|l_i)}{\sum_{i=1}^{T} p(v_j|l_i)}$
end for
//derive the the location of the mobile device using MAP estimation
$\hat{l} = \text{argmax}_{l_i}\left(\prod_{j=1}^{D} p(l_i|v_j)\right)$
return $\hat{l}$
```

posterior probability $p(l_i|v_j)$ for AP j could be obtained by

$$p(l_i|v_j) = \frac{p(l_i)p(v_j|l_i)}{\sum_{i=1}^{T} p(l_i)p(v_j|l_i)}, \tag{2.15}$$

where $p(l_i)$ is the prior probability for the device to be placed at position l_i. Generally, $p(l_i)$ is assumed to have a uniform distribution. Therefore, the posterior probability $p(l_i|v_j)$ is obtained in the step 9. Also we assume that the posterior probability $p(l_i|v_j)$ is independent for each AP; hence we derive the the location of the mobile device using MAP estimation (step 12).

2.4 Experimental Study

2.4.1 Experiment Configuration

We implement DeepMap system with commodity WiFi devices to evaluate its localization performance. Gaussian process for indoor localization is leveraged as benchmark [43] in this section. To guarantee the fairness, both schemes are applied with the save datasets, the Broun Hall dataset and the public dataset. The training data and test data for both schemes are identical. The same online localization algorithm presented in Section 2.3.4 is also used in both schemes to ensure the fairness. First, the DeepMap performance is evaluated with the Broun

Hall dataset, which is collected from the third floor of Broun Hall in the Auburn University. In this scenario, we use Wi-Fi Scanner 3.4 to collect all RSS measurements in both offline stage and online stage. The surveillance area is about 2300 m^2. As seen in Fig. 2.6, 157 locations is covered in the training data, which are represented by blue dots. The space between blue dots is 2m. The test data are gathered from 43 locations, which is represented by yellow squares. The space between yellow squares is 4m. In this dataset, the RSS values are gathered from 433 APs, which consists of both 5GHz APs and 2.4GHz APs from various manufacturers. Furthermore, the RSS values for unavailable APs are assigned to -99 dBm as discussed.

We also deploy our DeepMap system on the public dataset to examine its performance. The area is 860 m^2 approximately, which includes eight classrooms, four offices and a main hallway [44]. Also, all RSS values in the dataset are extracted from both 5GHz APs and 2.4GHz APs. The training data are collected from 82 locations and the test dataset includes RSS values from 34 locations. The distance between two adjacent locations is 2.6 m. The RSS values for online localization phase are collected from each testing location twice, each of which faces a different direction.

2.4.2 Accuracy of Location Estimation

First, we check the localization accuracy with the adequate training data. Fig. 2.7 illustrates the cumulative distribution function (CDF) of localization errors for the proposed DeepMap and Gaussian Process. In both schemes, all fingerprints collected from Broun Hall are leveraged to train the models. For the DeepMap, the median localization error is about 1.3m. However, Gaussian process obtains the median error of about 1.5m. The comparison shows that DeepMap has a better accuracy than Gaussian process. In addition, only 60% localization errors for Gaussian process could be lower than 2m, while 75% localization errors reach the same level with DeepMap. We also find that the largest error for Gaussian processes is 6.182 m which is greater than the largest errors for DeepMap, 5.207 m. Thus, DeepMap exceeds Gaussian processes in localization accuracy when adequate fingerprints are available.

Similarly, Fig. 2.9 shows the localization performance of both schemes with the Public dataset. When all training dataset is leveraged to train these two algorithms, the median errors

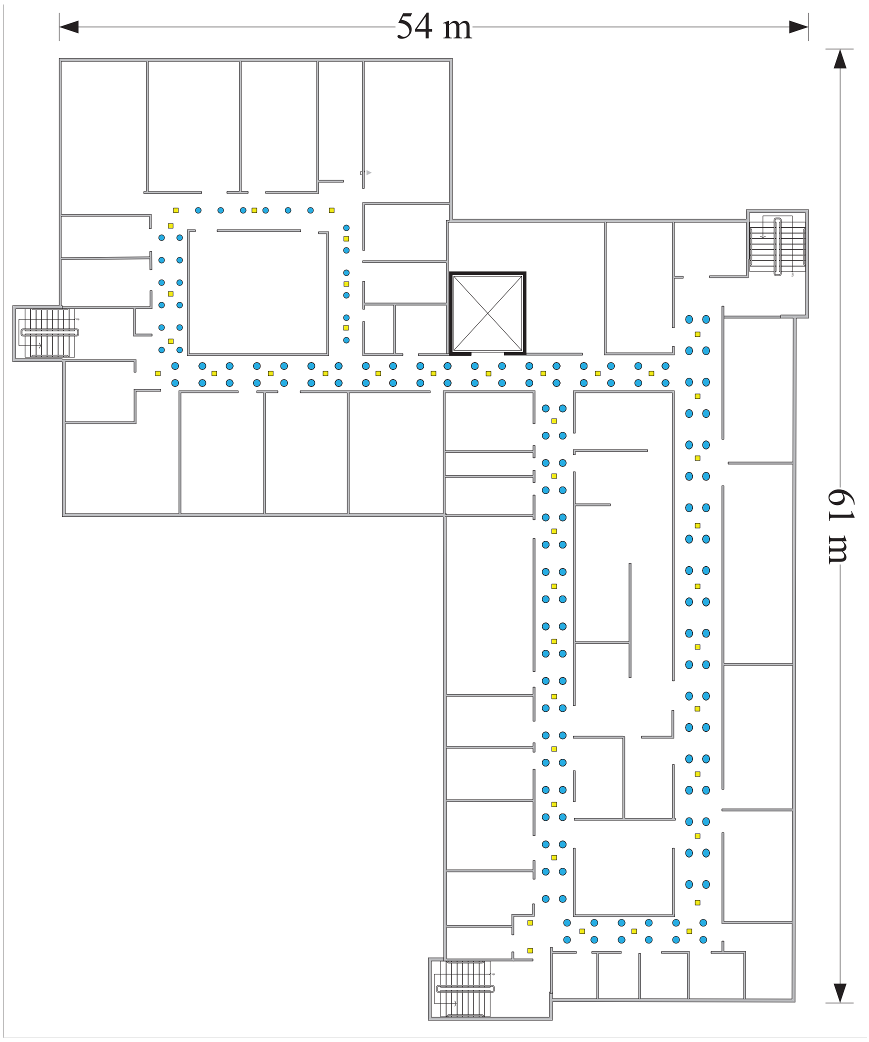


Figure 2.6: Layout of the third floor of Broun Hall at Auburn University: training locations are marked as blue dots and testing locations are marked as yellow squares.

for DeepMap and Gaussian process are 1.668 m and 2.2017 m, respectively, which proves DeepMap shows a better performance in localization than Gaussian process. For DeepMap, we also notice that more than 80% errors are under 2.8 m. However, only 65% test points for Gaussian process could reach the same level. Thus, DeepMap outperforms Gaussian process based on this public dataset, when the whole training dataset is available.

We also evaluate the performance of both scheme with deficient training data. With Borun Hall dataset, the mean distance error is 1.569 m when all fingerprints are leveraged. However, when 90% training data is used by DeepMap, the minimum mean distance error is reached, which is 1.536 m. Besides, we also find that the distance errors are robust to the percentage of fingerprints when more than 50% fingerprints are available to DeepMap. Gaussian process

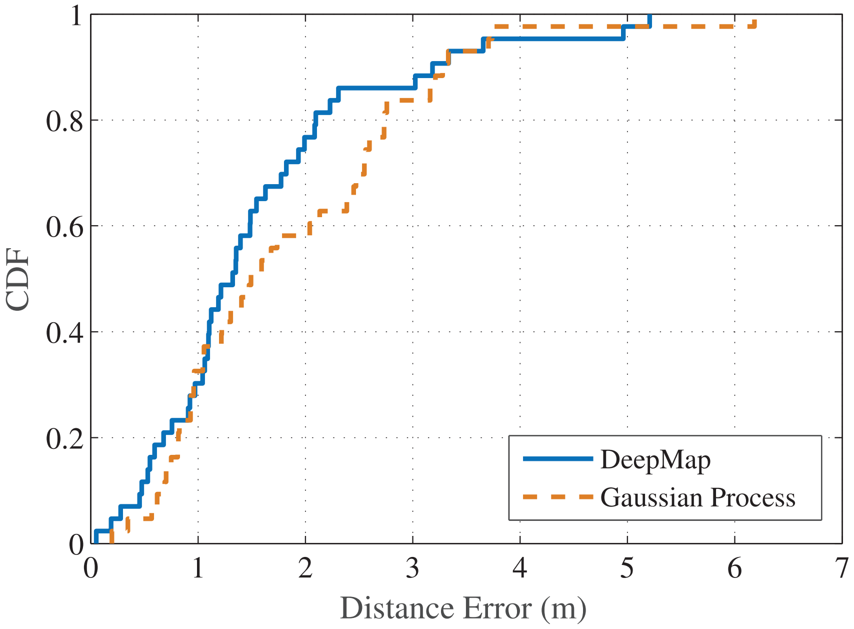


Figure 2.7: CDF of localization errors for the proposed DeepMap and Gaussian Process approaches using 100% Broun Hall dataset.

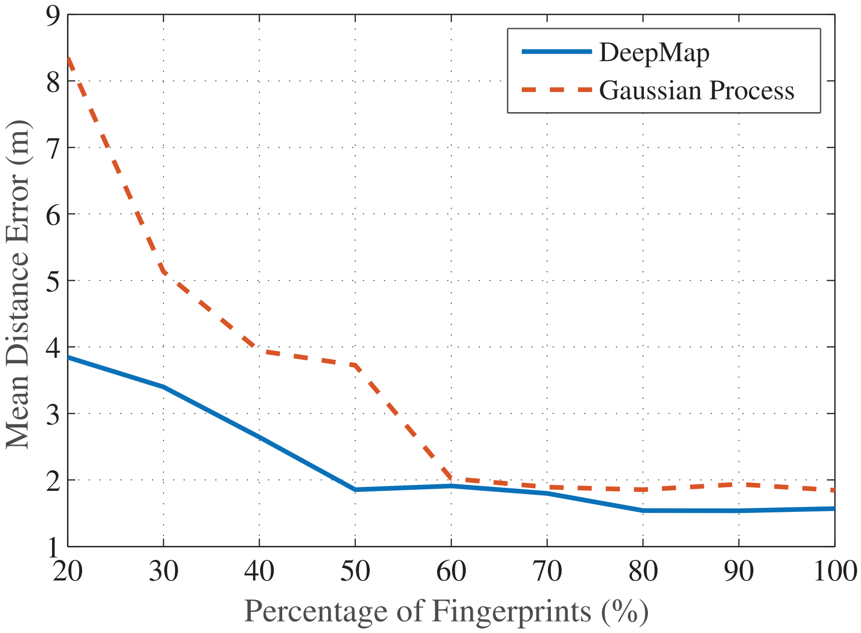


Figure 2.8: Mean localization errors for the proposed DeepMap and Gaussian Process approaches using different percentages of fingerprints in the Broun Hall dataset.

achieves the best performance, 1.845m, when all training data are avaiable. However, it is still greater than the lowest mean distance error for DeepMap. Even though Gaussian process also shows robustness in mean distance errors when more than 60% fingerprints are available, the distance error downgrades dramatically to 3.725m when 50% of fingerprints are leveraged in

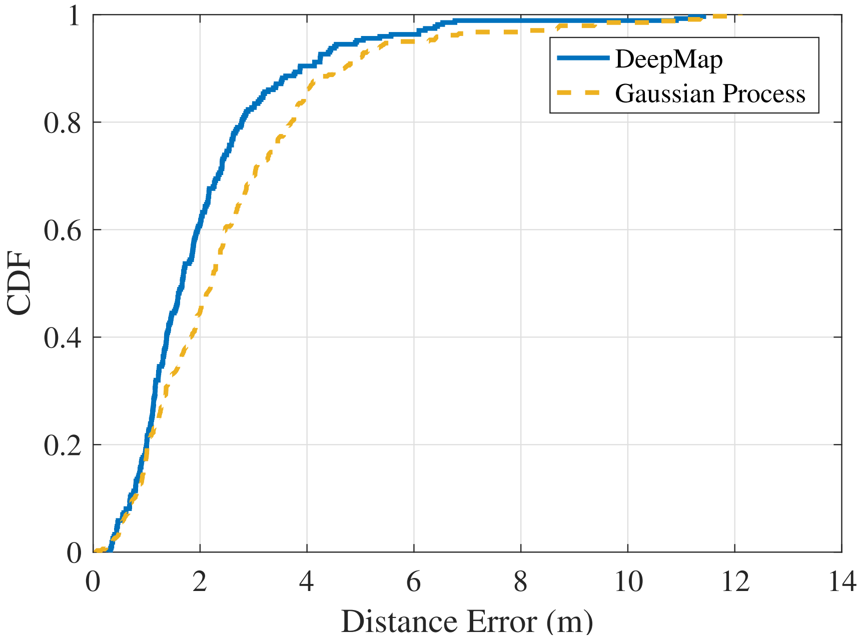


Figure 2.9: CDF of localization errors for the proposed DeepMap and Gaussian Process approaches using 100% public dataset.

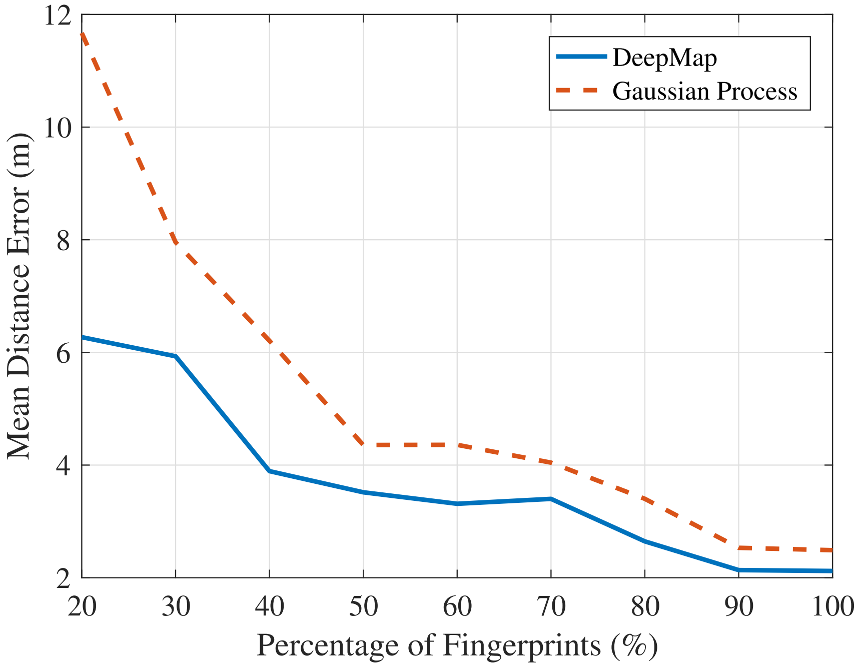


Figure 2.10: Mean localization errors for the proposed DeepMap and Gaussian Process approaches using different percentages of fingerprints in the public dataset.

Gaussian Process. Then, the mean distance error for Gaussian process explosively increases to 8.3496m when 20% of fingerprints are used to train the model. However, the distance error for DeepMap system does not change abruptly. The worst largest distance error for DeepMap is

3.8447m when 20% of fingerprints are utilized to train the model. In Fig. 2.8 , we also note that all errors obtained by DeepMap are always lower than the errors achieved by Gaussian process.

Furthermore, we investigate the effect of deficient data to localization errors with the public dataset. In Fig. 2.10, the minimum distance errors with 2.12 m for DeepMap and with 2.489 m for Gaussian process are obtained when all training data are available. However, the distance error for Gaussian process increases dramatically with the decrease of the training data. When only 20% training data are available for Gaussian process, the maximum distance error is 11.67 m. Both methods show larger errors when the algorithms are trained with only 20% training data even though the maximum error for DeepMap is about the half of the maximum error for Gaussian process. However, the performance for the proposed DeepMap is improved significantly, when 40% datasets are leveraged to train the algorithms. The mean distance error for DeepMap is 3.892 m, which approximates to the performance of Gaussian process when 70% public dataset are used. Thus, DeepMap shows a more robust performance with a smaller training dataset.

In conclusion, when the training data is adequate for training DeepMap and Gaussian process, both of the schemes are able to regress the outline of the RSS surface. However, comparing with the Gaussian process, a more detailed map could be generated by DeepMap, which leads to a more accurate localization precision. For the map constructed by Gaussian process, it does not contains much detail, such as non-stationary components, thus the minimum error is slightly greater than DeepMap. When only partial fingerprints are available, the localization error of Gaussian process increases dramatically, while DeepMap exhibits the robustness to the inadequate fingerprints. The nonlinear characteristics are captured by DeepMap even with a few number of fingerprints, thus achieving a higher localization accuracy.

2.4.3 Impact of Various System Parameters

To investigate the impact of system parameters on the localization precision of our DeepMap system, all of the fingerprints in the Broun Hall dataset are leveraged in the following experiments. In each experiment, the training process repeats 5 times with identical parameter settings. The average test result is recorded as the final result.

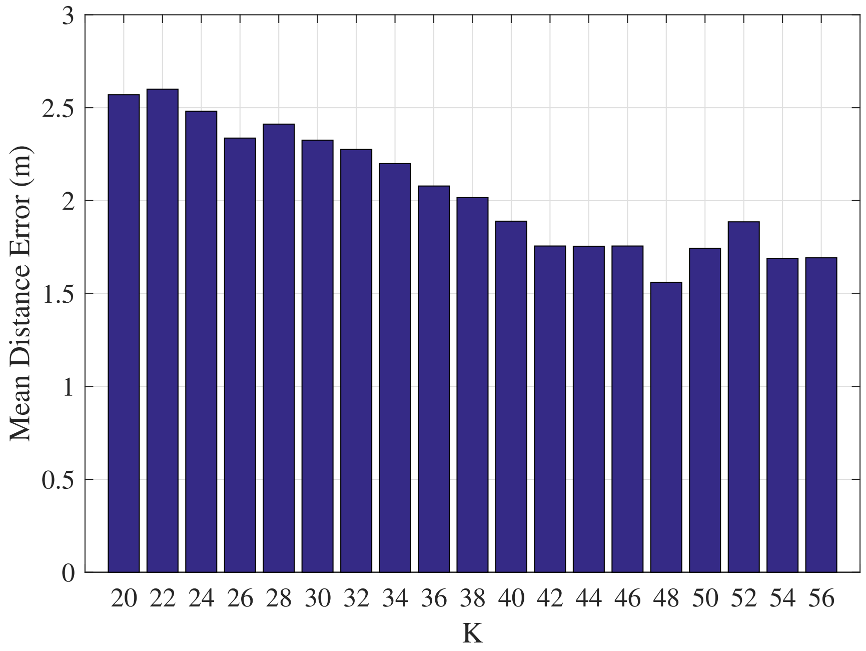


Figure 2.11: Mean distance errors with different values of K.

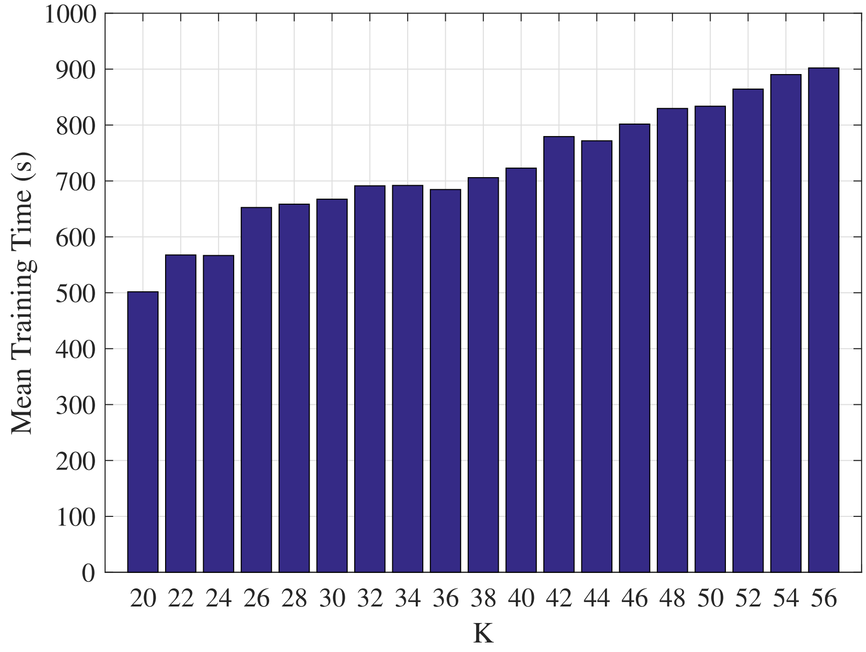


Figure 2.12: Mean training times with different values of K.

Impact of the number of the inducing points

In the DeepMap system, K represents the number of inducing points. Even though it could be different for every layer of the overall structure, we keep the numbers of inducing points the same in each layer to simplify the study. As is shown in Fig. 2.11, we compare the mean

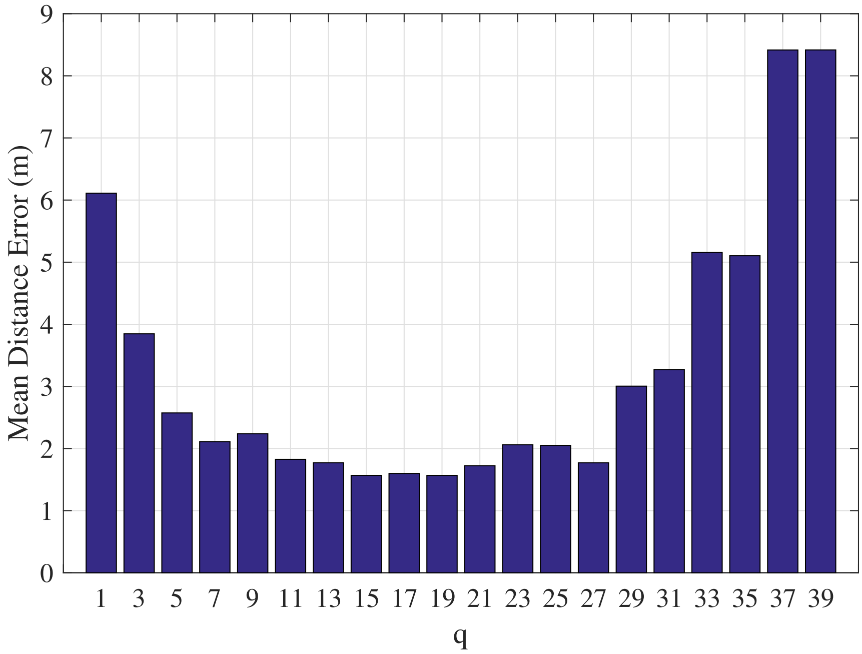


Figure 2.13: Mean distance errors with different values of q.

distance errors with the different values of K. According to Fig. 2.11, the mean distance error decreases gradually with the increment of the value of K. After K is greater than 40, the impact of K to the mean distance error reduces and the mean distance error converges to about $1.65m$. Fig. 2.12 depicts the corresponding training times for different values of K. As we can see, the mean training time goes up with the increase of K. Considering that the training time would not jeopardize the user experience in the online stage, K is set to 48 for obtaining the best localization performance in the following experiments.

Impact of the number of latent node

q appears as the number of latent nodes in the deep Gaussian process. Ideally, each latent node could have its own weight w_q. but the weight could also be removed by setting t zero. We design a specific experiment to study the effect of q to the performance of our DeepMap system and to optimize the value of q to achieve the best localization precision. In this experiment, the value of K is set to 48 to eliminate the effect of K. 20 different values of q are introduced to the DeepMap system to evaluate their effect on the performance of our system. For each q, the training process is repeated 5 times to avoid the randomness of the results.

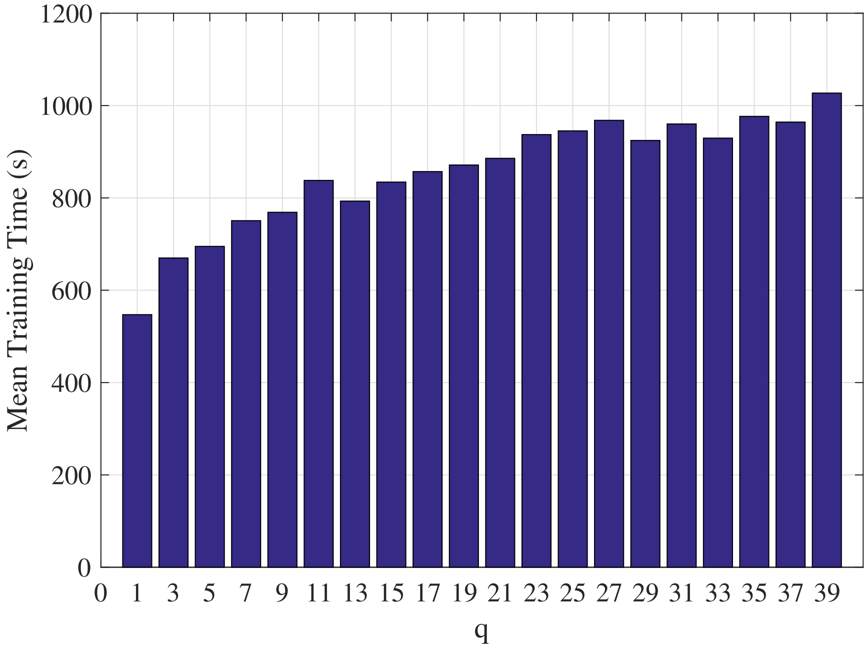


Figure 2.14: Mean training times with different values of q.

Fig. 2.13 depicts the mean distance errors for increased q. As the number of latent nodes raises from 1 to 9, the mean distance error declines from about $6m$ to about $2m$ rapidly. When the value of q is in the range between 11 and 27, the mean distance error does not fluctuate significantly. The lowest error happens when the value of q is in the range from 15 to 19. As soon as q is greater than 27, it produces a sharp rise in the mean distance errors. As we can see, the mean distance error increases from $1.77m$ to $8.4m$. Therefore, we conclude that the localization precision of our DeepMap system could be degraded by an oversized q, even though the weight for the corresponding latent node could be eliminated. We also investigate the impact of q to the mean training time. Similar with the impact of K to the mean training time, the mean training time goes up gradually with the increasing q. To obtain the best localization precision, the values of q is set to 17 in the following experiments. Fig. 2.14 illustrates that the training time is only about 14 minutes when q is 17. It is noteworthy that all of the fingerprints in the Broun Hall dataset are leveraged in this experiment. The training process would speed up if fewer fingerprints are utilized in the training process. Thus, the DeepMap system could react to the change of environment by updating the fingerprints and training the deep Gaussian process in real-time.

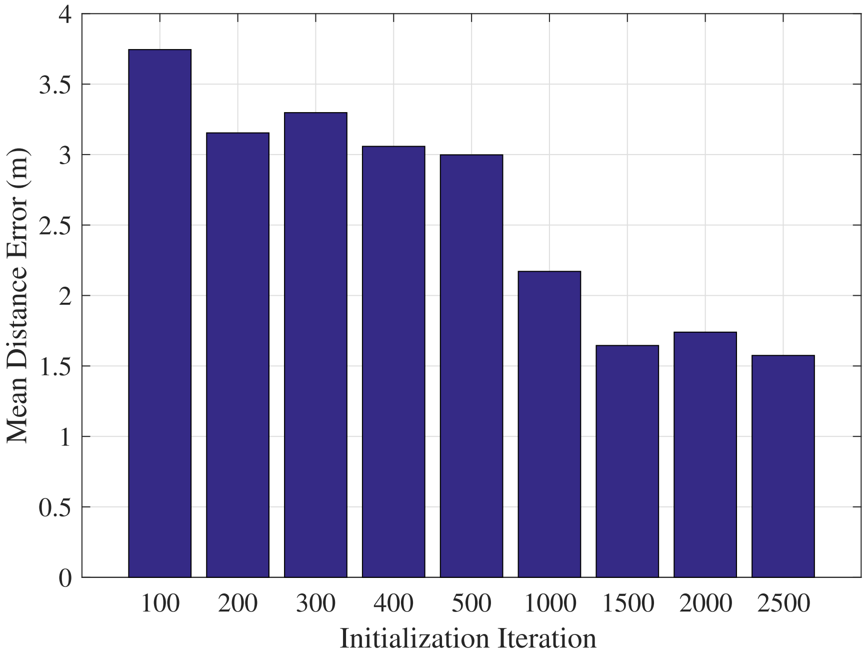


Figure 2.15: Mean distance errors with different numbers of iterations for initialising the variational distribution.

Impact of the number of the iteration for initialising the variational distribution

Fig. 2.15 plots the influence of the number of iterations performed for initializing the variational distribution on the localization precision of our system. As is shown in Fig. 2.15, the mean distance error drops slightly when the initialization iteration increases from 100 to 200. In the initialization iteration range between 200 and 500, the localization precision keeps stable and the mean distance error is about $3m$. To better look into the effect of the initialization iteration to the localization precision, the initialization iteration gap between the rest of the experiments is enlarged to 500. With 1000 initialization iterations, the localization precision improves significantly. When the initialization iteration reaches to 1500, the mean distance error continues to decrease. However, the localization performance of our DeepMap system does not keep enhancing, once the initialization iteration is greater than 1500. The mean distance error stays at the level of about $1.6m$.

Impact of the resolution of the constructed RSS radio map

Fig. 2.4 depicts the reconstructed RSS radio map, which is generated by 100% fingerprints in the BrounHall dataset. In the Figure, the green dots represent the reconstructed RSS values

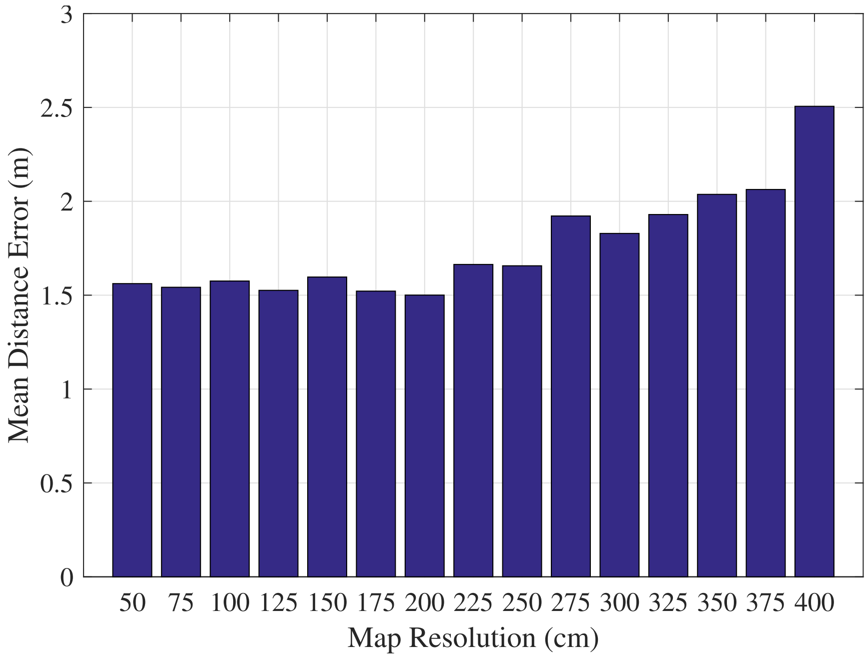


Figure 2.16: Mean distance errors with different map resolution.

at the reference positions. The resolution of the reconstructed RSS radio map is decided by the density of the reference points. To investigate the impact of the map resolution to the performance of our DeepMap system, 15 maps with the different resolutions are generated with the same well-trained deep Gaussian process. By observing from Fig. 2.16, we find that the localization precision is not affected by the map resolution significantly when the map resolution is lower than $200cm$. Also, with the information from the table 2.1, we notice that the size of the RSS radio map shrinks rapidly when the map resolution increases from $50cm$ to $200cm$. Thus, it is safe to say that a fine-grained RSS radio map is not essential to the better performance for the DeepMap system. However, the mean distance error goes up if the map resolution keeps increasing. The worst localization precision is obtained with the map resolution of $400cm$. According to the table 2.1, the map size and the time for construction map are related to the map resolution inversely. Even though the mean distance error is about $1.5m$ when the map resolution is $50cm$, the RSS radio map would be enormous, which costs 12.4 MB. Correspondingly, the testing time and the map construction time are also higher than the other results obtained by lower resolution maps. Combining the results in the table 2.1 and the mean distance error in 2.16, the best performance of DeepMap system is achieved when the resolution of the RSS radio map is set as $200cm$. With the resolution, the map construction

Table 2.1: Map Construction Time, Testing Time, and Map Sizes with Different Map Resolution

Map Resolution (cm)	50	75	100	125	150	175	200	225	250	275	300	325	350	375	400
Map Construction Time (s)	24.3	9.38	5.08	3.13	2.11	1.56	1.24	0.95	0.77	0.63	0.53	0.46	0.40	0.33	0.33
Testing Time (s)	0.91	0.46	0.27	0.20	0.15	0.12	0.11	0.09	0.09	0.08	0.07	0.04	0.04	0.03	0.03
Map Size (MB)	12.4	5.52	3.17	2.03	1.38	1.03	0.83	0.64	0.51	0.44	0.36	0.31	0.27	0.22	0.21

time and testing time reduces to 1.24 second and 0.11 second, respectively. With the help of this shorter testing time, DeepMap system has the potential to provide the real-time localization service. Also, because of the optimized resolution, the map size is only 0.83 MB, which is friendly to most mobile devices.

2.5 Related Work

Indoor localization has drawn great attention with the proliferation of mobile devices. Recently, variant indoor localization systems devote to promoting localization precision with advanced methods and algorithms. In this section, we review the fingerprinting base indoor localization system first. Then we mainly discuss two types of fingerpringitng based localization systems, which are closely related to our DeepMap system, i.e., deep learning based localization system, and radio map based localizaton system.

RADAR [25], the first Received Signal Strength(RSS)-based fingerprinting localization system, localizes the target by comparing the fingerprints collected in the online stage with the RSS fingerprints database with a deterministic method. To improve the localization precision, Horus leverages a K-nearest-neighbor based probabilistic method, achieving a mean accuracy of 0.6 meters. However, the nature of RSS restricts the performance of the RSS-based systems. Firstly, the RSS values are influenced by the multipath and shadow fading significantly. Thus, due to the diversity of RSS, two consecutive RSS readings, which are collected at the exact same location, could be different. Secondly, the RSS value is the coarse information obtained by averaging the amplitudes of all incoming signals. Comparing with the RSS, the channel state information(CSI) is more fine-grained, which depicts the characteristic of each subcarrier. FIFS [45] and PinLoc [46] utilize CSI to build fingerprints. The experimental results show that both FIFS and PinLoc outperforms Horus significantly in the same testbed. Although all these fingerprinting base systems perform agreeably in localization precision, the enormous fingerprinting database degrades their performance in mobile devices that have limited disk space.

Deep learning-based indoor localization systems rely on deep networks to extract features from CSI and leverage the features as fingerprints. DeepFi [47] is the first work to use autoencoder to extract features from CSI. It leverages the bias and weights from a well-trained three-layers autoencoder as fingerprints. PhaseFi [41] and DFLAR [48] propose to train the autoencoder with the phase values and images generated by CSI respectively. Also, WiDeep [49] improves the robustness of the localization by combining a stacked denoising autoencoders deep learning model and a probabilistic framework. Furthermore, [50, 51] contribute to device free indoor localization with deep auotoencoder networks. Due to the powerful abilities of the Convolutional Neural Network (CNN) in the fields, such as computer vision, it also has been used to promote the performance of the indoor localization system. A 6-layer CNN is employed in CiFi [30]. In contrast to previous fingerprinting based systems, the CiFi system would not use the fingerprinting database in the online stage. It only stores a set of weights and biases to achieve localization. Besides, [52] promotes the localization precision by preventing the overfitting problem with a limited training dataset. ResLoc [31] proposes to utilize a residual network to obtain submeter level accuracy with a single access point.

However, because of the nature of fingerprinting based systems, the localization problem is treated as a matching problem or multi-classification problem. Therefore, the density of fingerprints is highly related to the performance of the fingerprinting based localization system. To address such problem, Surecose [33] and [43] generates the radio map for an indoor environment with Gaussian process, which models the RSS values in a continuous space. With the advantages of the interpretable radio map, researchers propose the solution for some existing problem about fingerprinting based localization. For example, WinIPS [53] leverages the Gaussian Process Regression (GPR) with Polynomial Surface Fitting Mean to predict RSS on virtual reference points (VRPs). It overcomes the laborious fingerprint collection in the offline phase, and updates the radio map automatically in a dynamic environment. DncIPS [54] presents FWA-GPR algorithm, which is based on the Gaussian Process Regression (GPR) with a fireworks algorithm (FWA). It is also robust to the change of the environment. However, the location of APs is not essential in the DncIPS, which contributes to the improvement of the flexibility of this system. Even though both WinIPS and DncIPS solve the problem of fingerprints update in the dynamic environment, their localization precision could not be comparable to the other deep learning-based localization systems.

2.6 Conclusions

In this chapter, we presented DeepMap, a deep Gaussian process for indoor radio map construction and location estimation system. Comparing with the traditional Gaussian process for radio map construction, our DeepMap system consists of a two-layer deep Gaussian process model, which is able to extract nonlinear characteristics from fingerprints. We propose a Bayesian training method in the offline stage to optimize the model parameters and a Bayesian fusion algorithm in the online stage. Moreover, extensive experiments are conducted to evaluate the performance of our DeepMap system. The results indicate that the DeepMap system overperforms Gaussian process in both datasets, the Broun Hall dataset, and the public dataset. The DeepMap system also shows its robustness with the deficient training data.

Chapter 3

MapLoc: LSTM-based Indoor Location Estimation using Confidence Interval Maps

3.1 Introduction

Recently, With the rapid development of the Internet of Things (IoT), location based service (LBS) has drawn increasing attention from various fields, such as robotics, retailing, manufacturing, and smart buildings. Instead of using specifically designed sensors for location estimation, radio frequency (RF) signals, e.g., WiFi, have been a popular choice for indoor localization systems due to its wide deployment in indoor spaces. Fingerprinting is a popular indoor localization method, which generally consists of two stages: offline fingerprint collection and online location estimation. In the offline stage, fingerprints in the form of, e.g., WiFi received signal strength (RSS), are collected in the service area and labeled with the corresponding coordinates. Then, in the online stage, the unknown location of a mobile device will be estimated by matching the newly collected measurements with stored fingerprints. The performance of fingerprinting is thus largely affected by both the fingerprints and the matching method. Many prior works adopted various techniques in wireless communications, signal processing, and machine learning through these two aspects.

Various observations of RF signals have been utilized as fingerprints. For example, RSS was first used in [25]. Intuitively, RSS is negatively related to the distance between the transmitter and receiver. By using an empirical signal propagation model, the unknown location could be inferred roughly by triangulation. Even though RSS is resilient to slight environmental changes, it could not achieve fine-grained localization, especially when the number of APs is limited. For environments with rich AP resources, AP selection emerged to filter out the less useful RSS readings for boosted localization accuracy [55–57], which, however, is still an open problem. In addition, channel state information (CSI), as a fine-grained observation of the orthogonal frequency-division multiplexing (OFDM) physical layer (PHY), has been adopted as fingerprints in the past decade. It depicts how a signal propagates from the transmitter to the receiver through each subcarrier. Due to the nature of CSI, it is more sensitive than RSS to distance variations, and is also susceptible to the multipath effect and dynamic environments. Thus, various signal processing techniques have been proposed for eliminating the offsets introduced by the environment and hardware to enhance the quality of CSI fingerprints [29]. The extra cost of signal processing may impede the prevalence of CSI-based localization systems in mobile devices with limited hardware resources. Meanwhile, with the popularity of smart devices, increasing types of signals, such as light and earth magnetic field intensity, have been introduced as fingerprints [58]. It has been shown that such multi-modal fingerprints are complementary to each other and can help to make the system more robust.

In addition to the quality, the density of fingerprints is also a key factor that affects the accuracy of fingerprinting. To achieve high location accuracy, a site survey is needed to collection fingerprints at densely marked locations, which is usually time-consuming and laborious. Furthermore, such dense fingerprints are costly to update when the service environment is changed (i.e., change of furniture placement). As a result, there is a trade-off between the location estimation accuracy and system deployment cost, which needs to be carefully balanced when designing a fingerprinting system.

Another crucial factor to the success of fingerprinting is an effective and efficient location estimation (i.e., matching) method. In recent indoor localization systems, machine learning has been widely used as classifiers to estimate unknown locations in the online stage, such as K-Nearest Neighbors (KNN), support vector machines (SVM), and random forest [25, 59, 60]. Recently, deep learning models, such as multilayer perceptrons (MLP), convolutional neural networks (CNN), and recurrent neural networks (RNN), have been adopted for effective multiclass classification [30, 31, 47, 61]. However, such methods are still focused on solving the traditional fingerprint matching problem, which partitions the continuous service area into a discrete grid and is treated as a multiclass classification problem. This approach introduces a built-in error, even though the error can be mitigated by probabilistic methods [47, 62].

In this paper, we propose MapLoc, an indoor fingerprinting system that utilizes Deep Gaussian Process (DGP) to regress uncertainty maps and incorporates a Long Short-term Memory (LSTM) based method for location estimation. From the perspective of fingerprint quality, both WiFi RSS and earth magnetic field intensity are utilized as fingerprints in MapLoc. Since the magnetic sensors are available in many smart devices, the magnetic field intensity measurements are readily available. Moreover, MapLoc utilizes the inferred confidence intervals of the uncertainty maps to generate artificial trajectories of fingerprints, which are used in auxiliary learning to pre-train the location prediction model. By implementing a stacked LSTM network as a backend, we design a location prediction model for regressing the signal maps. And the estimated location will be inferred directly by the model. More specifically, a DGP is first implemented for uncertainty estimation in the service area. Then the artificial signal measurements are generated by sampling the distribution described with uncertainties. In addition, geometry constraints and user movement patterns are considered in trajectory generation. The generated signal measurements are used to compose signal sequences that supervise the pre-training of the location prediction model. To better regress the signal strength, an auxiliary loss is adopted in the training. Both location prediction and fingerprint estimation are used to calculate the loss for weight updating. Finally, the pre-trained model is fine-tuned with real signal sequence collected in the field. Fine-tuning forces the location prediction model to converge to the real signal surface, thus eliminating the cumulative error of the DGP model. In the online stage, the location of the target mobile device is readily predicted by the location prediction model using its newly measured signals and past trajectory in a small sliding window.

The main contributions of this paper are summarized below.

- An innovative localization framework is proposed by leveraging the uncertainty estimation capability of DGP. Continuous uncertainty maps are created by DGP using fingerprints measured at gridpoint locations. The fingerprints are then augmented by sampling the distribution described by the uncertainty maps. The generated signal measurements reflect their own stability, allowing deep learning models to learn the reliability of signals and select the effective measurements for location estimation.

- By introducing geometric constraints of the service area and user movement trajectories, the continuous nature of human mobility and the historical locations of the target device within a small window are taken into account. Furthermore, fingerprinting is no longer treated as a classification problem here. Rather, the location prediction model readily produces the estimated location in the manner of regression, thus mitigating the built-in error of the traditional approach.
- We leverage auxiliary learning in training the location prediction model. By introducing the signal measurement loss as one of the components of the auxiliary loss in supervise training, the LSTM-based location prediction model will be forced to learn the inherent relationship in the sequences of measurements. Compared with the traditional training approach that only uses isolated location as labels, signal sequences include much more features to guide and accelerate the training process.
- Multimodal maps, created using WiFi RSS and earth magnetic field strengths, are utilized in the MapLoc system. Such measurements are widely available and do not increase the cost and affect the compatibility of the system. It is easy to extend the proposed framework to include more types of measurements, such as light intensity, for future improved performance.
- We verified the performance of the proposed MapLoc system with extensive experiments in two representative indoor environments. The results demonstrate that MapLoc advances the the accuracy of location estimation by taking advantage of the uncertainty estimation provided by DGP and the bi-modal fingerprints.

In the remainder of this paper, we present an overview of related work in Section 4.2. The preliminaries and motivations are provided in Section 4.3. Section 4.4 presents the system design. In Section 3.5, we evaluate our prototype system, and in Section 3.6, we wrap up this paper.

3.2 Related Work

With the rise of the wireless communication [63–68], indoor location-based services have drawn a lot of attention from both academia and industry, due to their high social and economic value. Unlike outdoor localization systems, such as the Global Positioning System (GPS), which rely on the line-of-sight (LOS) reception of satellite signals, the performance of indoor localization is hampered by scattered and reflected signals due to the clutter environment. Indoor localization is still an open problem without a universal solution, despite a variety of techniques have been proposed in the literature.

3.2.1 Fingerprinting Approaches

Because of their adaptability and adequate accuracy, fingerprinting methods are commonly used in localization systems. The features derived from the observations are adopted for pattern matching in fingerprinting. RADAR [25] was one of the first attempts to use RF signals, where RSS was used as fingerprints. Aside from RSS, various types of observations were leveraged in prior works as well. CSI is a fine-grained observation from the PHY layer, which includes the amplitude and phase of each subcarrier of the OFDM PHY. FILA [69] demonstrated that CSI helps to improve localization accuracy and reduce latency. The quality of fingerprints, which can be viewed as a discrete radio map, plays a critical role in such systems. A basic and effective way to improve the quality of the radio map is to increase the number of fingerprints. However, collecting fingerprints is usually time-consuming and laborious, and in some cases, impossible. To minimize such effort, prior works [70–72] utilized Unmanned Aerial Vehicles (UAV) to replace manual labor. DeepMap [62] constructed a radio map with DGP using only a limited number of fingerprints. WiGAN [73] generated fingerprints for an unknown area with Gaussian Process Regression conditioned least-squares Generative Adversarial Networks (GPR-GANs). The authors in [74, 75] investigated the radio map adaptation and update problem to avoid the cumbersome recollection of fingerprints in dynamic environments. On the other hand, the quality of fingerprints keeps improving with the advance of technology, hence the evolution of radio maps. Gu et al. [76] eliminated multipath interference in WiFi signals with the Sparsity

Rank Singular Value Decomposition (SRSVD) method. Luo et al. [77] extracted nonlinear features from RSS signals by implementing Kernel Principal Component Analysis (KPCA). Furthermore, deep learning techniques have achieved an exceptional performance in feature extraction as well. To extract nonlinear features from observations, deep autoencoders were incorporated in [29,41,47,78], while [58,79,80] leveraged LSTM and its variants to evaluate the correlation between received RF signals for optimizing the fingerprints. In [30,31,61,81], CNN was used to extract fingerprints from multidimensional signal arrays for improved localization accuracy.

3.2.2 Geometry-based Approaches

In addition to fingerprinting methods, geometric methods, such as multilateration and triangulation, are widely used in indoor localization systems by exploiting the measurements for fine-grained information. Among various measurements, Angle of Arrival (AoA) is commonly employed in radar and acoustics systems. ArrayTrack [82] proposed a multipath suppression algorithm for eliminating the reflection paths between transmitter and receiver. SparseTag [17] proposed to use a spatial smoothing based method, which processed a sparse RFID tag array and decreased the angle estimation error to $1.831°$. Time of Arrival (ToA) based systems estimate the transmitter-receiver distance by measuring the traveling time of the signal. However, such systems require tightly synchronized clocks at the transmitter and receiver. Kang et al. [83] mitigated the time synchronization error and the NLOS error by introducing an iterative Time-of-arrival (iToA) algorithm incorporating a multivariate linear model. Also, Yuan et al. [84] proposed a unified factor graph-based framework for ToA based localization in wireless sensor networks. The framework provided a unified treatment of the inaccurate positions of transmitters and the asynchronous network. Even though the localization accuracy keeps increasing with these approaches, their performance is still insufficient for practical indoor services because of the required LOS signals and multipath-free environments.

3.2.3 Other Approaches

In addition to RF signal-based techniques, vision-based techniques are also popular with the emerging of robotics, autonomous vehicles, and Augmented Reality (AR) [85]. The localization algorithms rely on the inputs from sensors, such as RGB-D cameras and infrared cameras, to extract location information. The vision based techniques usually achieve centimeter level accuracy in real-time, outperforming most of RF signal based techniques. For example, MonoSLAM [86] is the first study to apply the simultaneous localization and mapping (SLAM) approach with a single uncontrolled camera, with centimeter level accuracy at 30Hz real-time performance. AprilTag [87] created a visual fiducial system that enables full six degrees-of-freedom (6DOF) localization with a single image by using a 2D barcode tag as landmark. However, the computational cost of vision-based approaches constraints their deployment on IoT devices with limited computation power and short battery life [88]. Moreover, the visibility, occlusion, and privacy related issues further constrain the usage of vision-based approaches.

Indoor localization also takes advantage of the development of visible light communications (VLC). By analyzing the modulated light signal transmitted in the form of visible LED lights, many VLC signal-based localization techniques have been proposed. Because the diffused components emerging from multipath scattering are substantially weaker than the LOS component, the VLC-based localization system has a superior accuracy over RF signal based system, which usually suffer from strong multipath interference [89–91].

Acoustic signals have also been employed in localization systems. It provides precise localization at a low cost due to readily accessible equipment such as speakers and microphones, as well as excellent time-domain resolutions. For instance, EchoTrack [92] tracked hand trajectory with a built-in speaker array and microphone on smart phones by leveraging two-channel chirps to remove the multipath noise. Location estimation is enhanced by using the Doppler shift compensation and roughness penalty smoothing method. Vernier [93] achieved accurate motion tracking accuracy of less than 4 mm, by proposing a differentiated window based phase change calculation (DW-PC) to minimize the computation overhead for real-time tracking.

3.3 Preliminaries and Motivations

Gaussian process has been successfully applied for solving regression and probabilistic classification problems. A Gaussian process is described with its covariance matrix and mean function. Since the prediction is also Gaussian, confidence intervals can be estimated to depict the uncertainty of data distributed over a continuous space. Thus, a normalized signal strength map for a service area can be conveniently reconstructed with measured signal strengths and the corresponding coordinates by a Gaussian process regression model, which is give by

$$r(c) = f(c) + \epsilon, \tag{3.1}$$

where $r(c)$ and $f(c)$ represent the received signal strength and ideal signal strength for location c, respectively, and ϵ is the observation noise, which follows an i.i.d. (independent identically distributed) Gaussian distribution with zero mean and variance θ_n^2.

It is intuitive to assume that the received signal strengths r_i and r_j at coordinates c_i and c_j, respectively, also follow a joint Gaussian distribution with covariance $k(c_i, c_j)$, which is usually described using a kernel function as

$$k(c_i, c_j) = \phi^2 \exp\left(-\frac{1}{2l^2}\left|c_i - c_j\right|^2\right), \tag{3.2}$$

where ϕ and l are the hyper-parameters for depicting the signal variance and the smoothness of the kernel function, both of which can be estimated by using a maximum likelihood approximation method. Then the joint distribution of the estimated signal strength f_* of location c_* and the measured signal strengths $\mathbf{r}$ can be depicted as follows.

$$\begin{pmatrix} \mathbf{r} \\ f_* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix}\right). \tag{3.3}$$

The signal strength f_* can be inferred from the measured signal strength $\mathbf{r}$ by

$$\Pr(f_*|c_*, \mathbf{c}, \mathbf{r}) = \mathcal{N}(f_*|\mu_*, \Sigma_*) \tag{3.4}$$

$$u_* = \mathbf{K}_*^T(\mathbf{K} + \theta_n^2\mathbf{I})^{-1}\mathbf{r} \tag{3.5}$$

$$\Sigma_* = \mathbf{K}_{**} - \mathbf{K}_*^T(\mathbf{K} + \theta_n^2\mathbf{I})^{-1}\mathbf{K}_*, \tag{3.6}$$

where $\mathbf{c} \in \mathbb{R}^{N\times 2}$, $\mathbf{r} \in \mathbb{R}^N$, $\mathbf{K}_{**} = [k(c_*, c_*)]$, N is the number of positions where the measurements were taken, $\mathbf{K}$ is the covariance matrix of $\mathbf{c}$ with dimension $N \times N$, and $\mathbf{K}_*$ is an $N \times 1$ matrix of covariances between $\mathbf{c}$ and c_*.

Inspired by the Gaussian process based works, the DGP is leveraged in this paper to enhance the precision of the constructed map by recovering the non-stationary components of signal measurements. In our prior work [62], a two-layer DGP model was leveraged to extract nonlinear characteristics from RSS samples and construct radio maps. Compared with Gaussian process, DGP is able to regress complex input data by taking advantage of the fusion of kernels. Fig. 3.1 is a graphical representation of a DGP, which consists of three layers of nodes, i.e., the parent nodes C, the leaf nodes R, and the latent nodes H, which include two sublayers H_1 and H_2 [94]. For a 2D map generation problem, C is the set of training coordinates with dimension $N \times 2$, R denotes a signal measurement matrix of $N \times S$, and $H \in \mathbb{R}^{N\times L_{sub}}$. Here, N, S and L_{sub} represent the number of measured coordinates, the number of sensors, and the number of the intermediate latent dimensions in the sublayers, respectively. Therefore, the generative process is given by

$$h_{nl}^1 = f_l^H(c_n) + \epsilon_{nl}^H,\ l = 1, 2, ..., L_1,\ c_n \in \mathbb{R}^2 \tag{3.7}$$

$$h_{nl}^2 = f_l^{H^*}(h_{nl}^1) + \epsilon_{nl}^{H^*},\ l = 1, 2, ..., L_2,\ h_{nl}^1 \in \mathbb{R}^{L_1} \tag{3.8}$$

$$r_{ns} = f_s^R(h_{nl}^2) + \epsilon_{ns}^R,\ s = 1, 2, ..., S,\ h_{nl}^2 \in \mathbb{R}^{L_2}, \tag{3.9}$$

where $f^H \sim GP(\mathbf{0}, k^H(C, C))$, $f^{H^*} \sim GP(\mathbf{0}, k^{H^*}(H_1, H_1))$, and $f^R \sim GP(\mathbf{0}, k^R(H_2, H_2))$ are Gaussian processes, which connects the latent nodes H with parent nodes C, themselves, and leaf nodes R, respectively. The automatic relevance determination (ARD) covariance functions

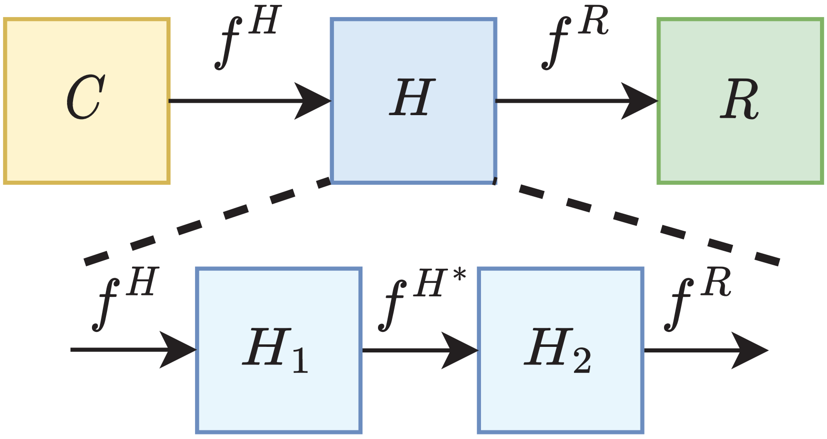


Figure 3.1: The DGP model for signal map construction.

for the Gaussian Processes is defined as

$$k_{ARD}(c_i, c_j) = \phi_{ARD}^2 \exp\left(-\frac{1}{2}\sum_{l=1}^{L} w_l (c_{i,l} - c_{j,l})^2\right), \tag{3.10}$$

where w_l is the weight for each latent dimension and ϕ_{ARD} is a hyper-parameter. For different inputs, the Gaussian processes, f^H and f^R, only be dependent on the covariance function k_{ARD}. To find the optimal hyper-parameters, Bayesian training is leveraged to maximize the marginal distribution of the observed signal measurement R, which is given by

$$\max \; \log p(R) = \log \int_{C,H} p(R|H)p(H|C)p(C). \tag{3.11}$$

The outstanding performance of DGP for generating a detail-rich signal map has been demonstrated in [62]. With the deep and heterogeneous nonlinear structure, the DGP handles the non-stationary components in complex signal measurements and extracts the detailed information about the distribution of real WiFi RSS measurements in indoor environments.

Despite the fact that the detailed maps created by DGP improves localization accuracy, the uncertainty information, which could also be retrieved using DGP, was largely ignored in our prior work [62]. Indeed, the uncertainty information just happens to be a convenient tool for evaluating the reliability of sampled signals. Fig. 3.2 illustrates a uncertainty radio map constructed by DGP using the measured RSS data from a specific AP in a public dataset [95].

The map includes three layers, a green layer representing the upper confidence bound of the map, a blue mean layer, and a peach layer denoting the lower confidence bound of the map. The confidence bound layers depict the 95% confidence interval of the signal distribution. The position of the AP is implied in Fig. 3.2. At the top-left corner of the map, the signals are the strongest and the most stable, because this area is close to the AP. When the distance is increased, the signal strength decreases and fluctuates more considerably. For the locations that are beyond the coverage of the AP, the signal strength drops to -100dBm and settles there. The RSS data from this AP, obviously, would be more constructive in locating target devices in the top-left region, while this AP would have a negative impact on locating targets in the map's central area because the RSS samples in the area would be highly random with large fluctuations. Such a pattern of uncertainty indicates that the signal stability varies depending on the location. And different patterns of uncertainty map would also be obtained for different APs. Thus, in MapLoc, we can sample the Gaussian distribution that is defined by the mean and confidence intervals in the uncertainty map to generate artificial measurements that depict the stability of the signal. The following LSTM-based location prediction model will exploit such fluctuations to distinguish the optimal signal measurements for location estimation. Moreover, Fig. 3.3 plots the uncertainty map generated by DGP using earth magnetic field observations. It follows a similar trend as in Fig. 3.2, in which the signal stability changes at different locations, and is complementary to the RSS uncertainty map. Both RSS and magnetic field data will be used in this effort to improve the accuracy of localization.

On the other hand, the proposed MapLoc system also takes into account the trajectory of the target device in a sliding time window. The trajectories can be reasonably synthesized by leveraging the movement pattern of target devices and geometry constraints (e.g., the shape of the room or corridor). Using the uncertainty maps, artificial signal sequences can be generated along such movement trajectories. The artificial signal sequences are used to pre-train the LSTM-based location prediction model, which is then fine-tuned with real collected signals in the field. The pre-training process guides the location prediction model by learning the signal reliability, while fine-tuning mitigates the cumulative error introduced by imprecise uncertainty maps.

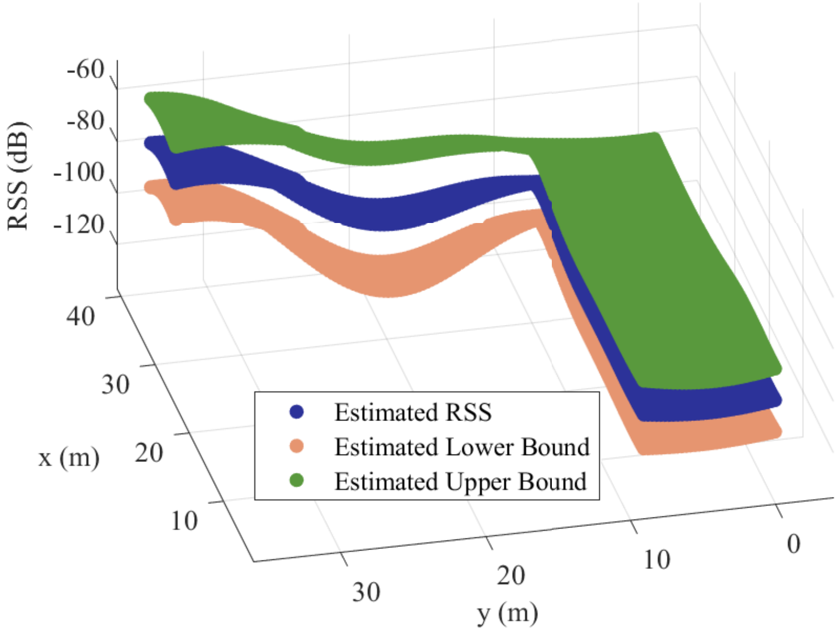


Figure 3.2: An RSS uncertainty map constructed by DGP.

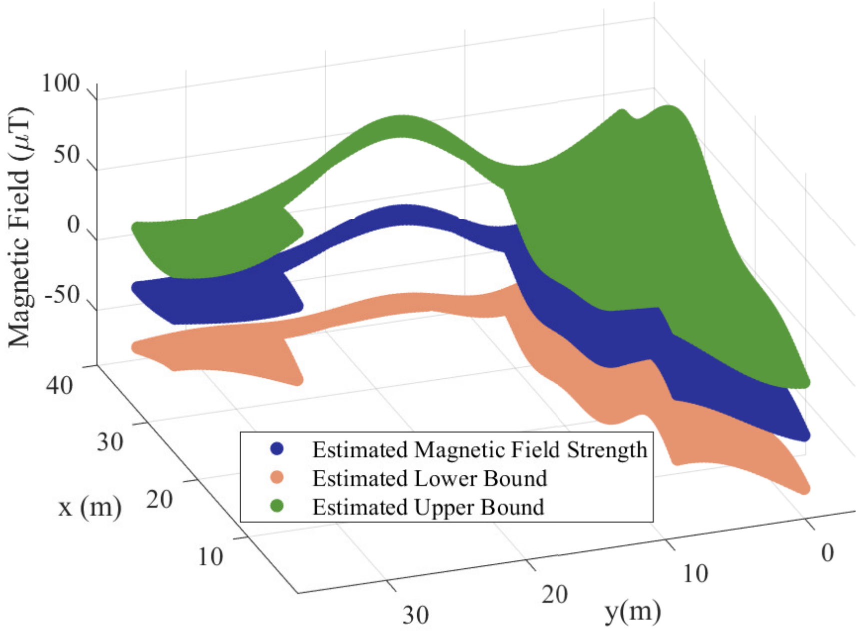


Figure 3.3: An earth magnetic field intensity uncertainty map constructed by DGP.

3.4 system overview

Fig. 3.4 presents the system architecture of the MapLoc system, where the green and blue blocks represent the components in the offline stage. More specifically, the green blocks are related to collecting signal measurements and their corresponding coordinates, whereas the

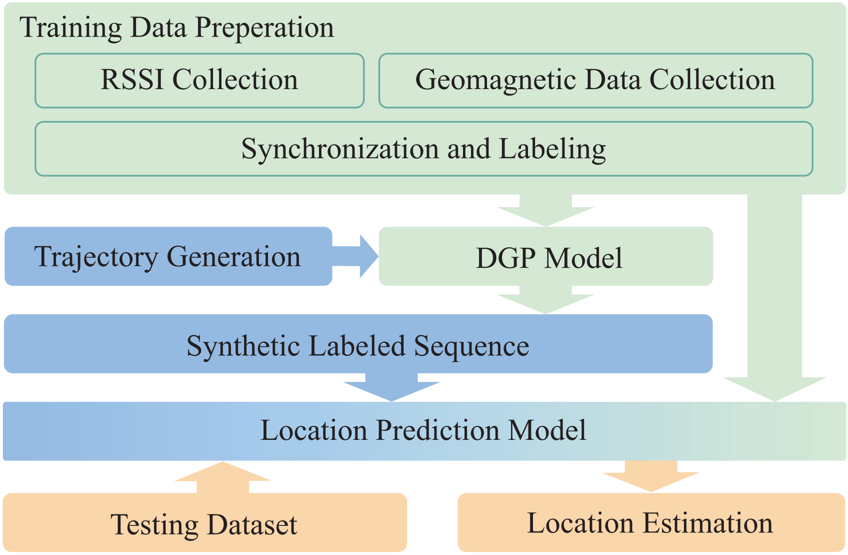

Training Data Preperation
RSSI Collection
Geomagnetic Data Collection
Synchronization and Labeling
Trajectory Generation
DGP Model
Synthetic Labeled Sequence
Location Prediction Model
Testing Dataset
Location Estimation

Figure 3.4: The MapLoc system architecture.

blue blocks are associated to the synthesized signal measurements and their coordinates. The location prediction model is unique in that it is pre-trained with the synthesized RF data and then fine-tuned with the collected RF data, which is why it is colored in gradients (from blue to green). The yellow blocks in Fig. 3.4 represent the components in the online stage.

Similar to traditional fingerprinting systems, MapLoc also consists of two stages: an offline stage for data collection and model training, and an online stage for location estimation. In the offline stage, WiFi RSS measurements as well as magnetic field readings are collected with the built-in sensors in the mobile device. The measurements comprising the collected bi-modal sequences, which are tagged with the corresponding coordinates where the data was measured. For each location, we collect RSS measurements from as many APs as possible. Since the set of visible APs usually varies from location to location, we force the RSS measurements from those inaccessible (i.e., out of coverage) APs to be -100 dBm to ensure consistency in measured data.

Localization with MapLoc includes two parts as well. The collected bi-modal signal measurements are first leveraged for training the DGP model to generate their uncertainty maps. The uncertainty map includes the mean value and the upper and lower bounds of the 95% confidence interval, as illustrated in Fig. 3.2. The uncertainty map will then be leveraged to synthesize artificial bi-model signal sequences for enhancing the training of the location prediction model, which is introduced to consider the trajectory (or, historical) information of the target device in location estimation. The model is first pre-trained with the artificial signal sequences synthesized by sampling the uncertainty maps, and then fine-tuned with the collected bi-modal sequences to avoid the cumulative errors introduced by the DGP model. In the online stage, the DGP model will not participate in location estimation. The estimated location will be obtained by combining the previous trajectory information with a time window W with the signal measurements from the current unknown location.

3.4.1 Offline Training

Offline training of the MapLoc system includes pre-training and fine-tuning. The DGP model is first trained using the bi-modal signals that have been collected. The location prediction model will first be trained using the artificial bi-modal sequences generated by the DGP model, and then fine-tuned using the signal sequences composed of collected signal measurements from the field to ensure that it converges to the real-world situation.

Pre-training

First, the collected signal measurements are used to train the DGP model. Because the DGP model focuses primarily on the signal distribution, the temporal information in the signal sequence is neglected during the training. To improve the structure of the DGP model and optimize the related hyper-parameters, a simple approach is employed to assess the quality of the uncertainty map generated by the DGP model. As shown in Algorithm 2, the constructed uncertainty map $\mathbf{M}$ is a $G \times S \times 3$ matrix, which includes an upper confidence layer, a mean layer, and a lower confidence layer. Here, G denotes the number of gridpoints in the map. It has to be $100,000$ to reach a resolution of 0.01 m for an area of 10 m^2. S represents the number of available signals. For example, we have $S = 10$ if the WiFi RSS measurements are collected from 7 APs, since each magnetic field reading is a vector with three elements (mag_x, mag_y, mag_z), describing the magnetic field intensity for the north, east, and vertical directions, respectively. The mean layer m is constructed to evaluate the overall quality of the

Algorithm 2 Pseudocode for measuring the quality of the uncertainty map

```
Input: the measured verification sample $r_j^k$ and the corresponding coordinate $c_*^k$, the mean layer of the uncertainty map $m_j$ for the $j$th signal, the number of gridpoints $G$ in $m_j$, the number of available signals $S$, and the number of verification samples $K$ ;
Output: the map quality $Q$ ;
 1: //$i$ represents the index of gridpoints in map $m_j$
 2: //$j$ denotes the index of signals
 3: //$k$ denotes the index of verification samples
 4: //$l$ denotes the coordinate of the gridpoints in map $m_j$
 5: for $k = 1 : K$ do
 6:     for $j = 1 : S$ do
 7:         for $i = 1 : G$ do
 8:             //compute the likelihood function $p(r_j^k|c_i)$
 9:             $p(r_j^k|c_i) = \exp\left(-\frac{1}{\lambda\sigma^2}\left\|r_j^k - m_j^{c_i}\right\|\right)$ ;
10:         end for
11:         //compute the posterior probability $p(l_i|r_j^k)$
12:         $p(c_i|r_j^k) = \frac{p(r_j^k|c_i)}{\sum_{d=1}^{G} p(r_j^k|c_d)}$ ;
13:     end for
14:     //use MAP estimation to infer location for the verification samples
15:     $\hat{c}^k = \mathrm{argmax}_{\{c_1,c_2,\ldots,c_G\}}\left(\prod_{j=1}^{S} p(c_i|r_j^k)\right)$ ;
16: end for
17: //compute map quality $Q$
18: $Q = \frac{1}{\exp(\frac{1}{2K}\sum_{k=1}^{K}(||c_*^k - \hat{c}^k||))}$ ;
19: return $Q$ ;
```

uncertainty map. K verification samples are collected from each gridpoint in the service area and labelled with the corresponding coordinates. We calculate the likelihood function $p(r_j^k|c_i)$ of the jth signal, which indicates the similarity between the kth verification sample r_j^k and the signal measurement at c_i in the uncertainty map m^j with a Gaussian kernel, as presented in Step 9. In MapLoc, the σ^2 and λ are set to 0.35 and 2, respectively. Thus, the posterior probability $p(c_i|r_j^k)$ is obtained conveniently by assuming the distribution over the G gridpoints is uniform (see Step 12). The coordinate estimation of the kth sample is given by choosing the gridpoint with the highest posterior probability. Eventually, the quality of the uncertainty map, Q, is evaluated based on the errors of the coordinate estimation in Step 18.

Based on the well-trained DGP model, a movement model is introduced to produce trajectories for generating artificial signal sequences. As shown in Algorithm 3, the stride length d is considered in the movement model and is restricted to 0.6 m. The azimuth γ is determined by

Algorithm 3 Pseudocode for artificial trajectory generation

```
Input: the length of the artificial trajectory $L$; the layout of the indoor environment $O$; the
    stride length $d$ ;
Output: the artificial trajectory $C$ ;
 1: //generate the coordinates $c_0$ randomly in the environment $O$ and initialize the trajectory $C$
 2: $C = \{randomPosition(O)\}$ ;
 3: while $C.length < L$ do
 4:     if $C.length == 1$ then
 5:         //$\gamma$ is a random initial azimuth
 6:         //generate the coordinate $c_*$ with the distance $d$ and the azimuth $\gamma$
 7:         //$c_{0_x}$ and $c_{0_y}$ are the x-axis and y-axis coordinates of $c_0$, respectively
 8:         $c_* = [c_{0_x} + d * cos(\gamma), c_{0_y} + d * sin(\gamma)], \gamma \sim U(-180°, 180°)$ ;
 9:     else
10:         //update $\gamma$ based on the previous azimuth
11:         $\gamma = \gamma + \gamma_t,\ \gamma_t \sim U(-40°, 40°)$ ;
12:         //$c_{-1}$ is the last coordinate in trajectory $C$
13:         $c_* = [c_{-1_x} + d * cos(\gamma), c_{-1_y} + d * sin(\gamma)]$ ;
14:     end if
15:     if $c_*$ in the environment $O$ then
16:         $C.append(c_*)$ ;
17:     end if
18: end while
19: return $C$ ;
```

the previous azimuth with a random offset between $-40°$ and $40°$. In Step 13, the coordinates in trajectory C are generated sequentially based on the previous azimuth. And the layout of the indoor environment is considered to eliminate the coordinates outside the service area (see Steps 15-17).

As shown in Fig. 3.5, the well-trained DGP model is utilized to generate the artificial signal r_N for coordinate c_N in trajectory C. According to trajectory C, the artificial signal sequences are assembled using the signal measurements generated by sampling the distribution $\mathcal{N}(\mu_N, \sigma_N^2)$ that is described by the mean μ_N and variance σ_N in the uncertainty map. It is noteworthy that the distribution is sampled M times to ensure that the generated signal measurements are able to represent the stability of signals. Furthermore, we employ a sliding window with a length of W for adjusting the size of the artificial sequences for training the LSTM based location prediction model. An artificial trajectory of length N will produce $N - W + 1$ training sequences. For each training sequence, the last signal measurement r_{i+W-1}^m and the corresponding coordinate c_{i+W-1} will be extracted as label for supervise training.

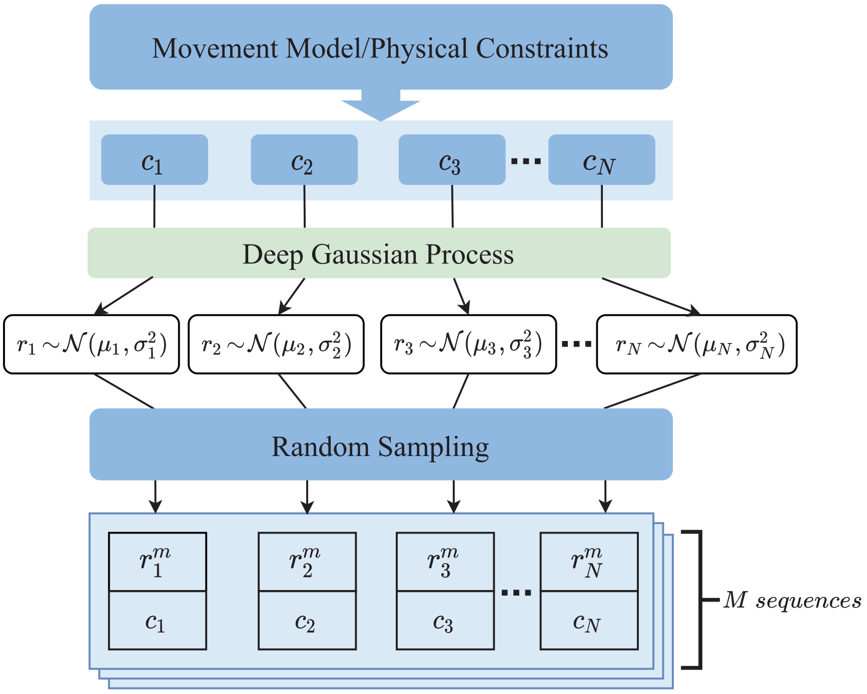

Movement Model/Physical Constraints
c_1
c_2
c_3
c_N
Deep Gaussian Process
$r_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$
$r_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$
$r_3 \sim \mathcal{N}(\mu_3, \sigma_3^2)$
$r_N \sim \mathcal{N}(\mu_N, \sigma_N^2)$
Random Sampling
r_1^m
r_2^m
r_3^m
r_N^m
c_1
c_2
c_3
c_N
M sequences

Figure 3.5: How to synthesize labeled signal sequences for pre-training the LSTM-based location prediction model.

The forward propagation of the location prediction model is depicted in Fig. 3.6. The backbone of the location prediction model is a stacked LSTM model, which is followed by a DNN for signal estimation (termed DNNS) and a DNN for location estimation (termed DNNL). To push the model to learn the signal map made by the DGP model and estimate location using the map, auxiliary loss is used in training. The signal values r_{i+W-1}^m in the label data is processed and concatenate with the output of the LSTM network in the DNNL model for predicting the unknown coordinate $\widehat{c}$. Then the MSE loss is calculated by comparing the label coordinate c_{i+W-1} and the location prediction $\widehat{c}$ by the DNNL. In parallel, a signal estimation $\widehat{r}$ is given by the DNNS using the output of the previous LSTM model as well. As a result, the loss function of the location prediction model is given by

$$\mathcal{L} = (1-\beta)\text{MSE}(r_{i+W-1}^m, \widehat{r}) + \beta\text{MSE}(c_{i+W-1}, \widehat{c}) \tag{3.12}$$

where β is a hyper parameter to adjust the influence of the two types of losses, while $\widehat{r}$ and $\widehat{c}$ are the predicted signal by DNNS and the predicted coordinate by DNNL, respectively.

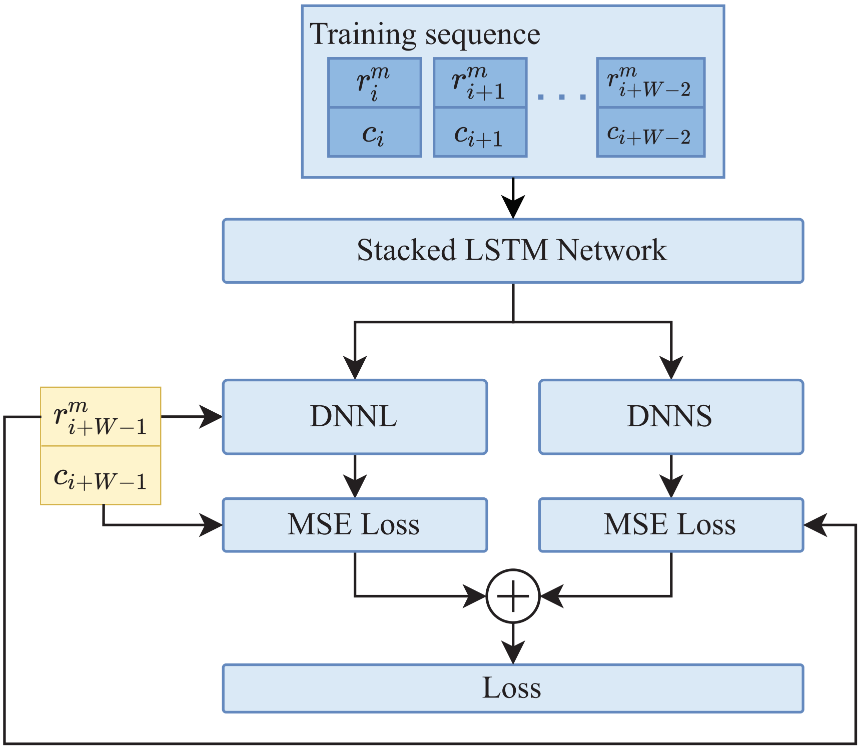


Figure 3.6: The LSTM-based location prediction model in MapLoc.

Fine-tuning

After pre-training, the location prediction model will be fine-tuned with collected bi-modal sequences from the service area. The collected bi-modal sequences, like the artificial sequences, are reorganized to form shorter training sequences using a sliding window of size W. The last bi-modal measurement of each training sequence is also used as the sequence's label to complete the supervised training of the model.

3.4.2 Online Testing

In the online stage, only the stacked LSTM network and DNNL will participate in location estimation. The location prediction model operates in a similar manner to autoregression models. The historical trajectory, including the received signal measurements and the corresponding coordinates, is fed into the stacked LSTM network. By combining the output of the LSTM network with the freshly collected signals from the current unknown location, the estimated location is deduced readily with the well-trained DNNL model. Because the localization problem is addressed as a regression problem in MapLoc, the built-in error associated with the discrete fingerprints can be avoided. Furthermore, since the estimated location is computed directly by the location prediction model, the cumbersome localization strategies used in prior work [62] are not needed anymore in MapLoc, which further reduces the computational cost, especially for mobile devices with limited computation resources and power supplies.

3.5 Experimental Study

3.5.1 Experiment Configuration

To demonstrate the performance of the MapLoc system, we evaluate it in two typical environments. First, we conduct experiments on the fourth floor of Broun Hall in the Auburn University Campus. In this scenario, we implement a prototype system using a Samsung Galaxy S7 Edge smartphone, which is equipped with a dedicated application for collecting magnetic field intensity data and WiFi RSS data simultaneously. As depicted in Fig. 3.7, the experiment covers an area of approximately 270 m^2. The black dots in Fig. 3.7 represent 255 sample locations (i.e., gridpoints) for training the DGP and the location prediction model. Except for some corner gridpoints, the distance between two adjacent training locations is 90 cm. 80 testing locations are randomly selected in the service area, which are not shown in Fig. 3.7. None of the testing locations overlap with a training location in this scenario. Moreover, RSS readings are collected from 224 APs, including all the available 2.4-GHz APs and 5-GHz APs from various manufacturers. To make the data size consistent, the RSS values of out-of-range APs are set to -100 dBm. The magnetic field strength is obtained from the on-device sensor directly, which is a vector including the magnetic field intensity for the north, east, and vertical directions.

The performance of the MapLoc system is also evaluated using a public dataset [95]. Fig. 3.8 plots the detailed floor plan where the public dataset was collected. The dataset covers a floor of 185.12 m^2, which includes three corridors and two offices. The fingerprints are captured from 325 gridpoint locations, shown as black dots in Fig. 3.8. The distance between two adjacent gridpoints is 60 cm. The data acquisition campaign was performed using a smartphone, SONY Xperia X2, and a smartwatch, LG W110G Watch R. We only utilize the data

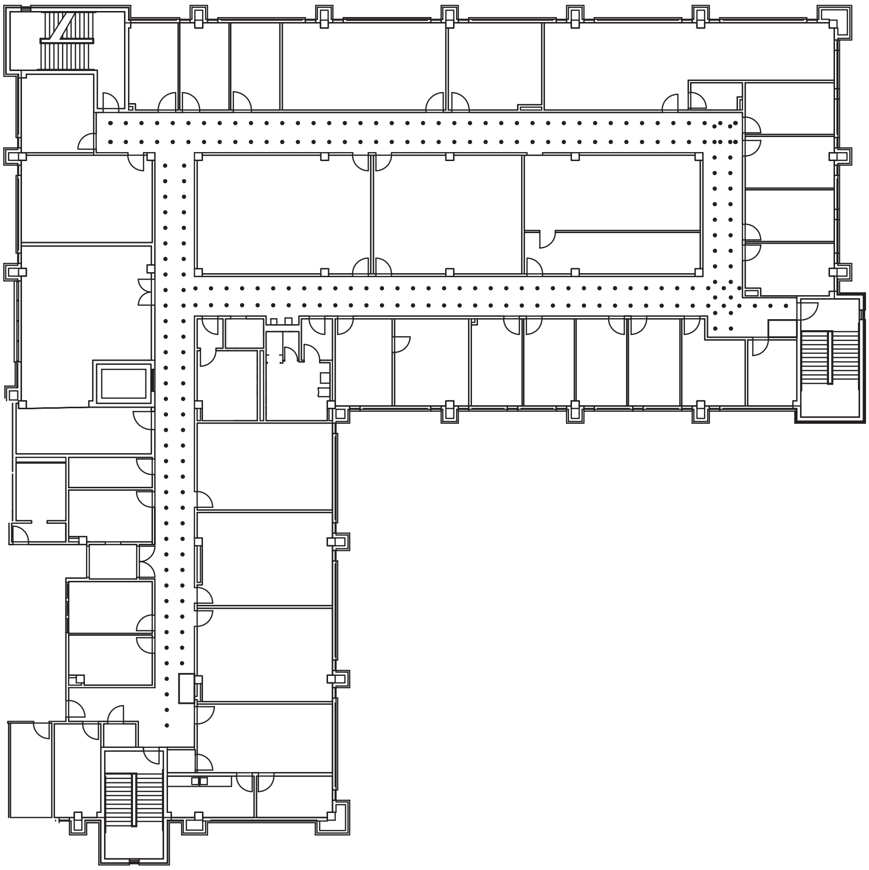


Figure 3.7: The floorplan for the Broun Hall dataset.

collected by the smartphone in this experimental study. The RSS data are captured from 132 unique APs, and the readings from an out-of-range AP are all set to -100 dBm. We only leverage 75 APs in the following experiments because some AP signals are very weak across the entire service area. Similar to the magnetic field intensity in the Broun Hall scenario, the magnetic field readings of this scenario are also vectors with three elements. Since the data acquisition campaign is conducted in this environment with the identical setting twice, we train and then test the MapLoc system using the datasets from different campaigns for a fair and realistic evaluation.

Identical settings of the location prediction model are deployed in both environments. Nine LSTMs are stacked one above another to form a stacked LSTM as backbone of the location prediction model. The number of features in the hidden state of LSTM is set to about 1.5 times of that of the input features, e.g., the number of features in the hidden state will be 150 if the number of available AP is 95. Each magnetic field reading is a vector of size 3×1 and the

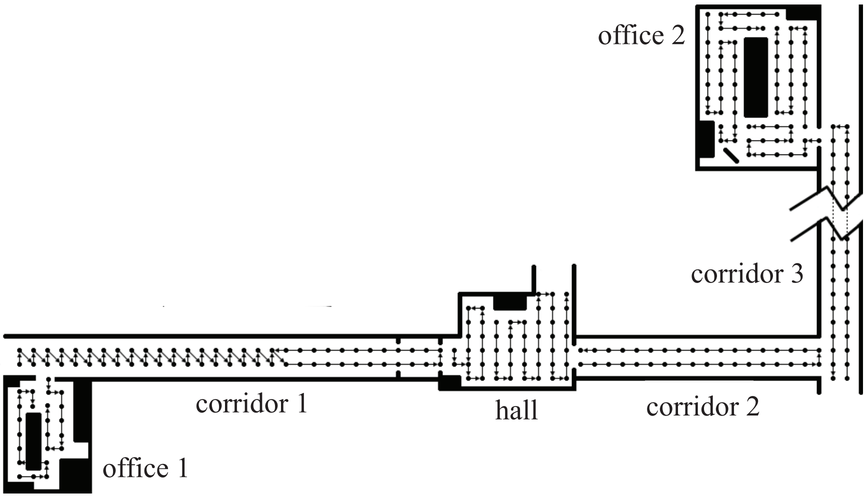


Figure 3.8: The floorplan where the public dataset was collected.

corresponding coordinates are in a 2D space. The hidden state of the last layer of the stacked LSTM is passed into the two DNNs for location estimation and signal estimation, respectively. DNNL is composed of 4 linear layers. The size of the input data r^m_{i+W-1} is first adjusted to 16 by a layer in DNNL, while the size of the hidden state from the LSTM is squeezed to 32 by another DNNL layer. By concatenating the outputs from the two layers, the estimated location is obtained by the remaining 2 layers in DNNL, where the output feature numbers of the layers are 16 and 2, respectively. The structure of DNNS is relatively simple. The hidden state from the LSTM is compressed by 3 linear layers in DNNS sequentially, where the output feature numbers of the layers are 256, 128, and the same as that of the input data r^m_{i+W-1}, respectively.

In both scenarios, the magnetic field intensity and WiFi RSS readings are min-max normalized. Considering that pedestrians usually do not make abrupt changes in their movements indoors, the stride length d is set to 0.6 m, and the azimuth offset γ_t is limited in the range between $-40°$ and $40°$. To accelerate the training process, a server with an Nvidia RTX 3090 GPU is leveraged for real-time trajectory generation and model training.

The following baselines are used in our comparison study:

- DeepMap: this is the scheme proposed in our prior work [62], where a Bayesian method is used for location estimation without using the uncertainty maps.

- LSTM: this scheme uses the same stacked LSTM network and DNNL to predict location, where the input is the trajectory and the corresponding RSS sequences of the target device. The LSTM model is trained with trajectory/RSS sequences sampled from the collected fingerprints.
- LSTM+DeepMap: the same location prediction model, shown in Fig. 3.6, is used to predict location, where the input is the trajectory/RSS sequences of the target device. The model is trained with sampled trajectory/RSS from the map created by DeepMap [62] without using the uncertainty maps.

3.5.2 Experimental Results and Analysis

Accuracy of Location Estimation

First, we evaluate the localization performance on the Broun Hall dataset. Fig. 3.9 illustrates the cumulative distribution functions (CDF) of localization errors for the proposed MapLoc system and the three baseline schemes. According to Fig. 3.9, it is obvious that MapLoc outperforms the other methods on the Broun Hall dataset. Despite the fact that both MapLoc and LSTM+DeepMap obtained a performance where 50% of the errors are less than 1 m, MapLoc has a distinct advantage that approximately 75% of location estimation have errors less than $1.35m$, whereas only 59% of location estimation obtained by LSTM+DeepMap accomplish the similar accuracy. This demonstrates the improvement brought about by the samples from uncertainty maps. In addition, Fig. 3.9 reveals the obvious deficiencies of LSTM and DeepMap in localization accuracy. The maximum localization error, 6.41 m, is from LSTM. The comparison demonstrates that the combination of LSTM and DeepMap contributes to higher precision localization. In MapLoc, the augmented training data produced by the DGP model benefits the location prediction model that uses LSTM as its backbone. By incorporating historical information into location estimation via the LSTM model, the localization accuracy of the DeepMap model is improved significantly as well. Based on the collaboration of DeepMap and LSTM, our proposed MapLoc successfully improves the location estimation accuracy by taking into account the uncertainties of different signals as well as historical information.

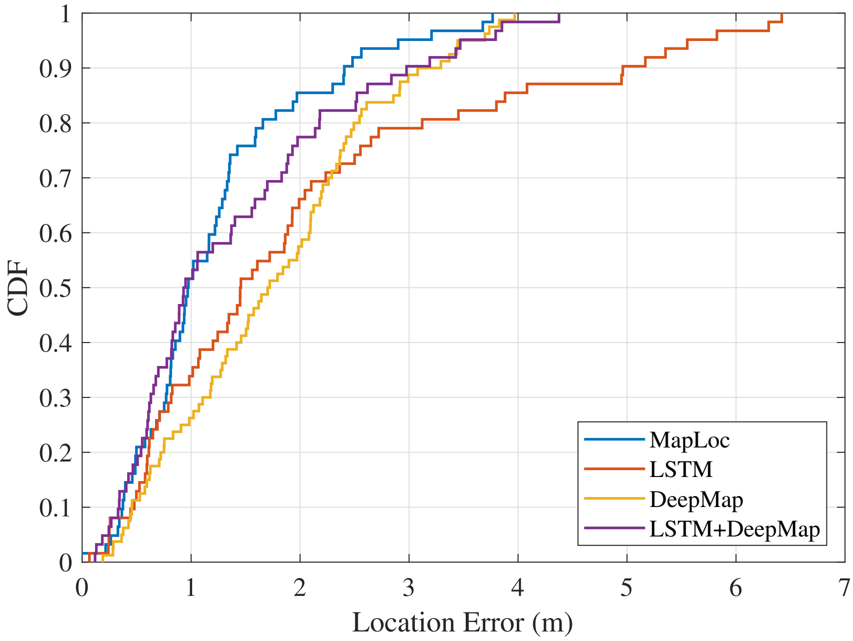


Figure 3.9: CDF of localization errors on the Broun Hall dataset.

We also conduct an experiment using the public dataset to investigate the performance of the proposed MapLoc system. The CDF of localization errors on the public dataset is displayed in Fig. 3.10. The results on the public dataset are similar to those with the Broun Hall dataset. MapLoc and LSTM+DeepMap keep the leading position in the comparison. Even though 50% of location estimation errors are lower than 1.1 m with both MapLoc and LSTM+DeepMap, the overall performance of MapLoc is superior to that of LSTM+DeepMap slightly. Because the artificial signal measurements are sampled from the uncertainty maps, the distribution of the generated measurements describes the measurement's quality. As a result, the location prediction model can learn the reliability of different types of signal measurement, the sets of measurements from different APs, and thus improve the accuracy of location estimation. Moreover, LSTM outperforms DeepMap in the public dataset scenario, although the maximum localization error, 14.26 m, is obtained with the LSTM method.

The main results in Fig. 3.9 and Fig. 3.10 are summarized in Fig. 3.11. The height of the bars represents *mean error*, whereas the black line in each bar represents *median error*. The location prediction model of MapLoc, denoted as LSTM in Fig. 3.11, and DeepMap, each only contribute to limited accuracy in location estimation. By combining these methods, the localization accuracy is increased significantly. The mean and median error on the public

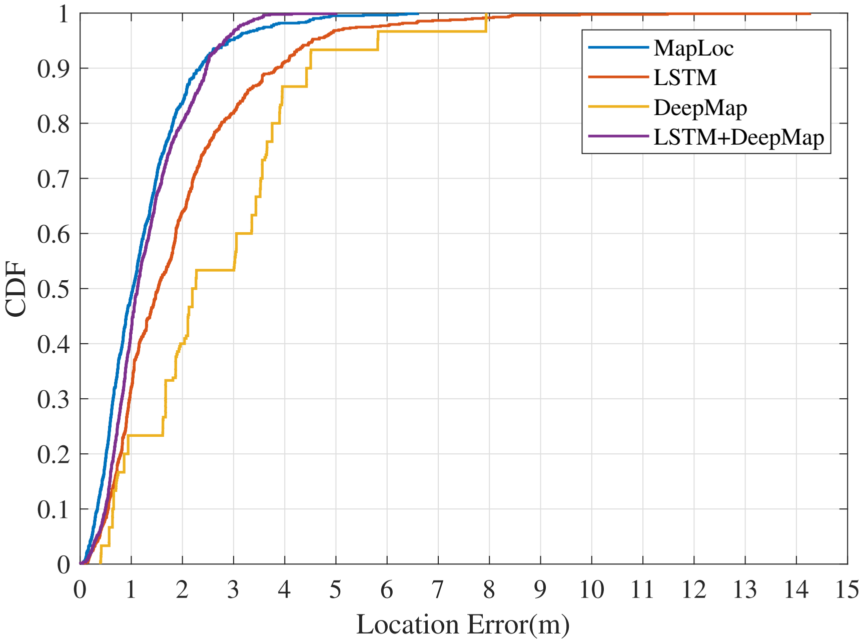


Figure 3.10: CDF of localization errors on the public dataset.

dataset reach 1.342 m and 1.145 m, respectively, while the mean and median error on the Broun Hall dataset are 1.374 m and 0.94 m, respectively. In MapLoc, the mean and median error are further reduced by augmenting the training dataset with the artificial data generated by sampling the uncertainty maps. In the public dataset scenario, the mean and median errors decrease to 1.234 m and 1.031 m, respectively. The mean error on the Broun Hall dataset reduces from 1.374 m to 1.211 m, whereas the median error reaches 0.9722 m.

Impact of Signal Selection

Previous results show that the Maploc system outperforms the systems that use LSTM and DGP separately. By leveraging the samples sampled from the uncertainty maps to measure the reliability of different signal sources (e.g., APs), the MapLoc system also beats the combination of LSTM and DGP. To investigate how the reliability of signal measurements affects MapLoc's location prediction and how the location prediction model contributes to better performance, we conduct experiments with both the Broun Hall dataset and the public dataset.

First, a random trajectory of each test datas zet is selected for the experiment. The corresponding signal distribution at the label coordinates is obtained with the DGP model in the MapLoc system. The mean and variance of signal measurements from all available sources,

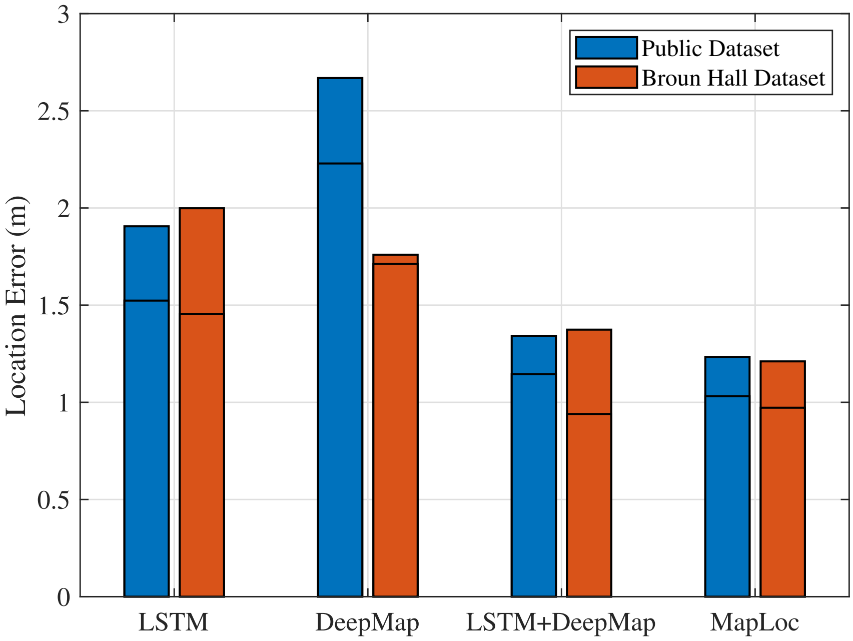


Figure 3.11: Mean and median localization errors on the Broun Hall dataset and the public dataset. The bars indicate mean error and the line within each bar indicates the corresponding median error.

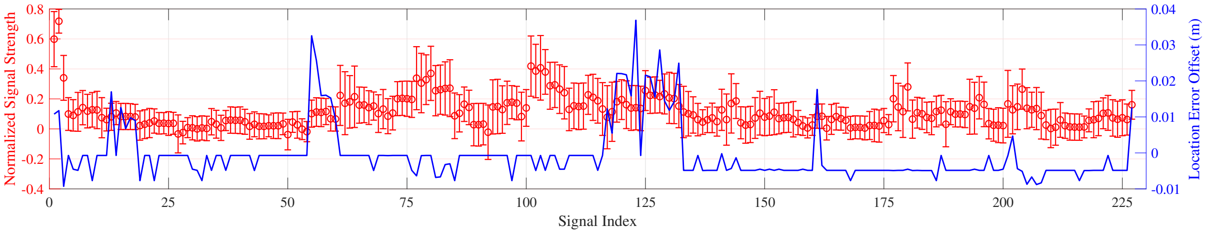


Figure 3.12: Explaining the importance of different signal measurements to the location prediction made by MapLoc system with the Broun hall dataset.

including magnetic field readings and WiFi RSSI, are represented by red circles and bars in Fig. 3.12 and Fig. 3.13. It is intuitive to suppose that a lower variance represents a trustworthy signal measurement, and signal measurements with higher mean values are more likely to influence the location prediction. To endow the MapLoc system with the ability to choose the signals intelligently, we sample the uncertainty maps to generate artificial signal measurements that describe its own reliability. We hope our location prediction model is able to learn how to recognize effective measurements from invalid and fluctuating signals. In the experiments, we double the signal measurements of the testing data in sequence to explore the performance of the location prediction model in signal selection.

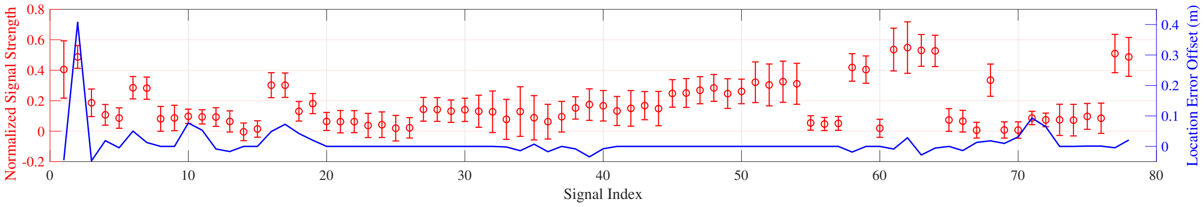


Figure 3.13: Explaining the importance of different signal measurements to the location prediction made by MapLoc system with the public dataset.

As previously stated, the signal measurements in the Broun Hall dataset contain 3 magnetic field components as well as RSSI readings from 224 WiFi APs. The experiment is repeated 227 times with the selected trajectory. In each repetition, we doubled a signal measurement individually. The blue line in Fig. 3.12 depicts the variations in distance errors caused by the doubled signal measurement. By comparing with the signal distribution denoted by the red circles and bars, it is clear that the location prediction model selects the optimal signal measurements, and the location estimation is more sensitive to changes in those measurements. As shown in Fig. 3.12, the first increment of the distance error happens at signal-2, which is the magnetic field reading's y-axis component. The signal measurement is much higher and more stable than the nearby signals. The next distance error fluctuation occurs between signal-13 and signal-18, where the mean signal values are higher than those on the right side and the signal variances are smaller than those on their left side. With the drop of the mean values of the signals, the location estimation of the MapLoc system is not influenced by the weak signals. The fluctuation in distance errors increase as the signals rise between signal-55 and signal-60, while the increase in distance errors disappears between signal-61 and signal-117. Even though the mean values of the signals between index-61 and index-117 are much higher than the rest of the signals in Fig. 3.12, the location prediction model detects the large variances of the signals, so the location estimation is not significantly affected by the signals. Another wild rise in location estimation is associated with signals near index-125, where the signals remain high and the variances remain stable. Furthermore, two distance error fluctuations occur at signal-161 and signal-227. It is clear that the signals remain stable, and they are stronger than the nearby signals.

Fig. 3.13 displays the signal measurements from the public dataset, which include RSSI readings from 75 WiFi APs and magnetic field components from 3 different directions. Because the number of signals in the public dataset is much smaller than that in the Broun Hall dataset, the location prediction model in the public dataset scenario is more sensitive to the doubled signal measurements. Fig. 3.13 shows the relationship between the distance error fluctuation and the signal stability. As we can see, the largest peak in Fig. 3.13 is related to the y-axis component of the magnetic field reading as well, because the signal component remains stable in a high level status. We also discover distance error fluctuations at signal-6, signal-17, and signal-62. Although these signals cause changes in the distance error with abrupt increases in signal strength, comparable changes could also be introduced by stable signals with lower signal strengths at index-10 and index-71. The location prediction model ignores changes in signal measurements between signal-20 and signal-60. Some signals in this range are stable, but the weak strength would not cause the degradation of the location estimation. Some signals are strong, but the location prediction model discards them due to their poor reliability.

Based on Fig. 3.12 and Fig. 3.13, we could conclude that the proposed location prediction model in the MapLoc system successfully extracts effective signal measurements from the weak and fluctuating signals by learning the artificial signal measurements that describe its own reliability. The selected signal is not only decided by the average signal strength but also determined by the stability.

Impact of System Parameters

In the MapLoc system, the auxiliary loss is used to force the location prediction model to acquire knowledge from the signal measurement generated by the DGP model and estimate the unknown location with the knowledge. β is introduced into the auxiliary loss function to balance the signal loss from DNNS and the location loss from DNNL. To optimize the accuracy of the Maploc system, we investigate the effect of β on the performance of the location estimation. Fig. 3.14 delineates the distance errors related to different values of β. In both scenarios, the accuracy of location estimation progresses when β is set as 0.8. Even though the mean distance error from the Broun Hall dataset is slightly decreased as β increases to 1.0, the

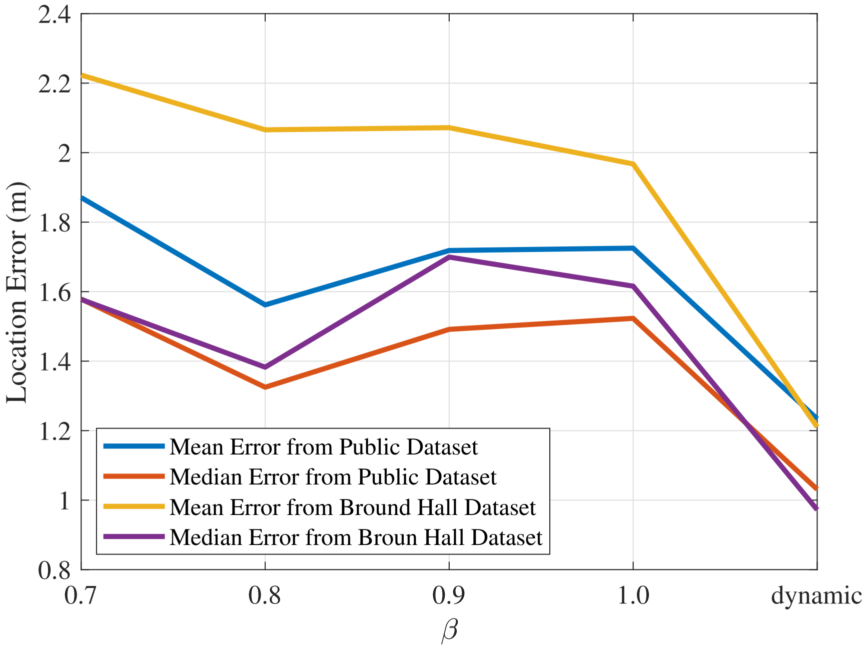


Figure 3.14: Mean localization errors for different values of β.

overall performance of the MapLoc system does not enhance significantly with the increment of β. Considering that signal estimation in the location prediction model is a supportive method for accurate location estimation, we adopted a dynamic way to adjust β based on the number of epochs. In the MapLoc system, the initial value of β is set as 0.6. When more than 200 epochs are completed, β updates every 100 epochs by decreasing 0.1. Eventually, the auxiliary loss would degenerate into a loss function determined by the location estimation error exclusively. Fig. 3.14 exhibits the performance promotion contributed by the dynamic β update.

Given that the position prediction model in the MapLoc system relies on the stacked LSTM network as its backbone, the window size W plays a crucial role in improving the accuracy of location prediction. Intuitively, a longer data sequence would contain more useful information to improve the precision of the location estimation; nevertheless, the longer sequence would incur additional system costs, such as an extra time cost in data collection. To study the effect of the sequence length on the accuracy of the location prediction, we conduct experiments with different window sizes on both the public and the Broun Hall dataset. Fig. 3.15 illustrates the distance errors resulted by different window sizes. Even though the distance error in the Broun Hall scenario is more sensitive to the change of the window size, the distance errors drop with the longer data sequence in both environments. When the window sizes are larger than 5,

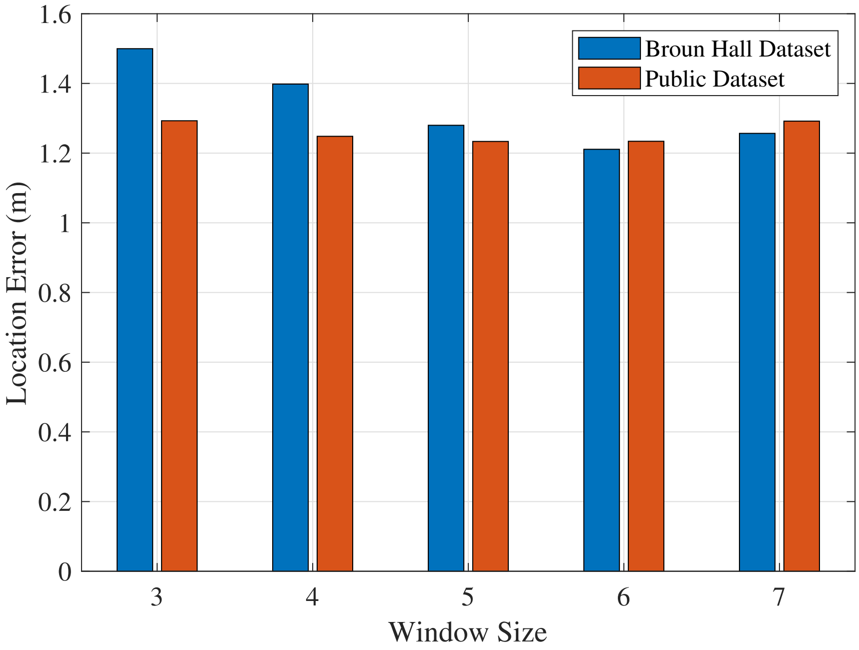


Figure 3.15: Localization errors effected by the size of the sliding window.

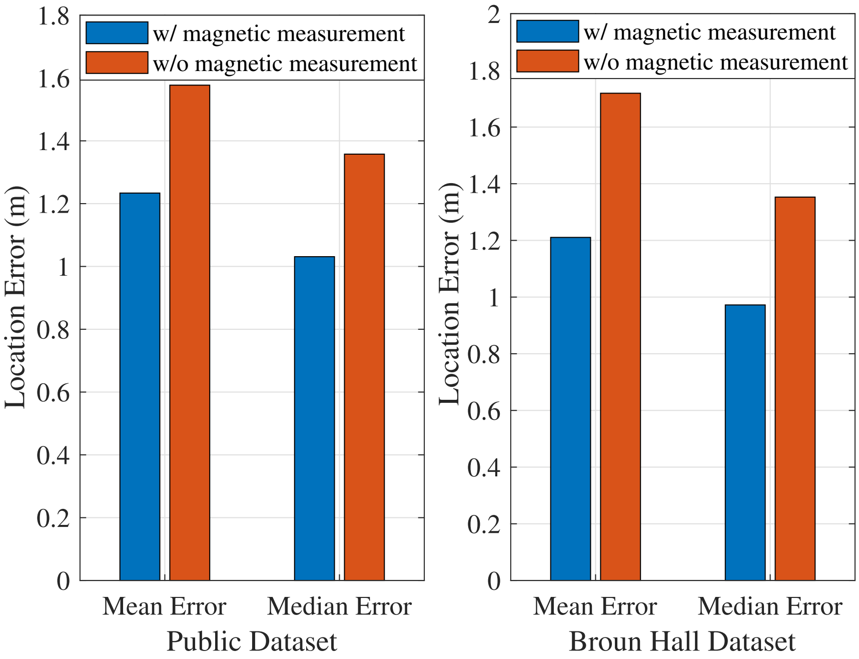


Figure 3.16: Reductions in localization error achieved by utilizing the earth magnetic field strength map.

the distance errors keep stable. The public dataset has the lowest location estimation error of $1.233m$ when the window size is 5, and the distance error is $1.234m$ when the window size is 6. Because we notice that the Broun Hall Dataset has the lowest distance error, 1.21m, when the window size is 6, we set the window size to 6 in the MapLoc system to ease system setup.

The MapLoc system uses multi-modal data to improve localization accuracy, as mentioned in previous sections. With the least amount of data processing, different types of signal measurements could be introduced into the system. In this prototype, magnetic field readings are used as a part of the MapLoc system's input data to improve the system's localization accuracy. Magnetic field components from different directions are treated as novel features of the input in our proposed MapLoc system after the max-min normalization. Fig. 3.16 illustrates the advancement brought by the bimodal data, which is composed of magnetic field measurements and WiFi RSSI measurements. The localization errors for the public dataset scenario decline notably. The mean error drops from $1.577m$ to $1.234m$ when the magnetic filed readings are taken into account, whereas the decline of median error reaches $0.327m$. A similar phenomenon happened to the broun hall dataset as well. Both mean error and median error are reduced remarkably. A huge depreciation of the mean distance error appears with the contribution of the magnetic field readings, where the mean distance error decreases from $1.719m$ to $1.211m$.

Because the location prediction model is pre-trained with artificial data generated by the uncertainty map, the DGP model is critical in the proposed MapLoc system. To assess the impact of DGP model parameters on the quality of the uncertainty map, Q, we investigate various combinations of latent dimension and number of inducing points in order to find the best parameters.

The latent nodes in MapLoc include two sublayers, H_1 and H_2. Fig. 3.17 and Fig. 3.18 show how the maps' quality Q is affected by the latent dimensions of the two sublayers, denoted by L_1 and L_2, respectively. The latent dimensions are tuned by gridpoint search in both scenarios. We first examine the effect of latent dimensions using the public dataset. Even though the quality of the uncertainty map increases with larger dimensions of the first layer when the second layer includes 6 or 8 latent dimensions, the relationship between the latent dimensions and map qualities is ambiguous. As shown in Fig. 3.17, the uncertainty map reaches the highest value when the latent dimension of the first layer is 7 and that of the second layer is 7. Also, two similar Q are achieved when the latent dimension of the first layer is 8. Because all the three Qs are close, there is no clear advantage to use different latent dimension settings. We try all

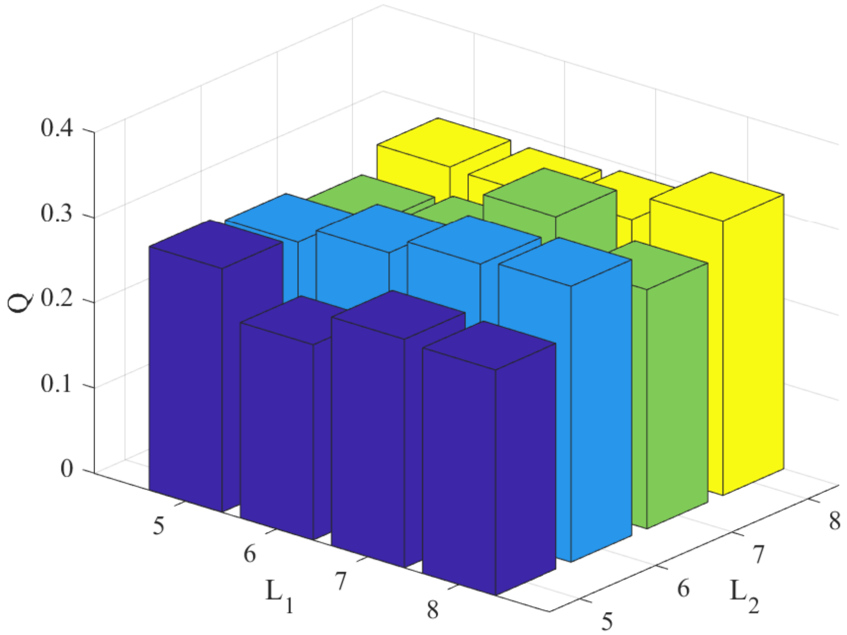


Figure 3.17: Q values versus the number of latent dimensions in public dataset scenario.

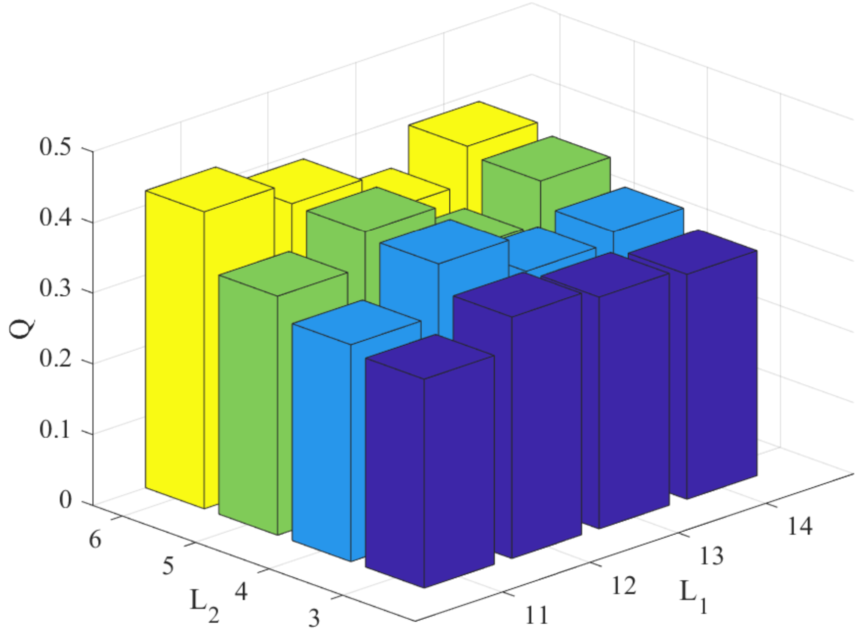


Figure 3.18: Q values versus the number of latent dimensions in Broun Hall dataset scenario.

three settings in training the location prediction model of MapLoc. Since the lowest validation error is reached when $L_1 = 8$ and $L_2 = 6$, we choose this setting for training the prototype of MapLoc using the public dataset. And the previous MapLoc results are all obtained under this setting.

Table 3.1: Q Values Affected by the Number of Inducing Points

Public Dataset								Broun Hall Dataset					
Inducing Points	11	10	9	8	7	6	5	Inducing Points	16	15	14	13	12
Q	0.248	0.317	0.274	0.275	0.267	0.266	0.240	Q	0.248	0.317	0.274	0.275	0.267

On the other hand, Fig. 3.18 reveals that the quality of uncertainty maps, Q, improves with increased latent dimensions of the second sublayer L_2 on the Broun Hall dataset. However, increasing the latent dimension of the first sublayer L_1 does not imply improved map quality. We find that increasing L_2 significantly improves the map quality Q when the first sublayer of the DGP model has 11 latent dimensions; whereas increasing L_1 does not contribute to further improvement of Q. According to Fig. 3.18, a gridpoint search yields the best map quality, i.e., $Q = 0.42$, for the Broun Hall dataset when $L_1 = 11$ and $L_2 = 6$.

Another key factor effecting the quality of the maps is the number of inducing points. For the DGP model of the MapLoc system, we choose identical numbers of inducing points for different layers to simplify the setting of the model. Similarly, we evaluate the effect of the number of inducing points on the quality of the uncertainty maps with both datasets. Table 3.1 presents the map quality Q obtained by different numbers of inducing points with the public dataset. According to the table, the worst map quality is acquired when each layer of DGP model only includes 5 inducing points. Along with the increasing number of inducing points, the map quality keeps enhancing. Even though the growth rate for the map quality is slow when the number of inducing points is between 6 and 9, a notable promotion is observed when 10 inducing points of each layer are involved in the training of the DGP model. Thus, the number of inducing points is set to 10 for the accurate location estimation in the public dataset scenario. The map quality stops improving as the number of inducing number reaches 11, where the map quality is close to that of the model with 5 inducing points in each layer.

In the Broun Hall dataset, Table 3.1 reveals a similar result regarding the number of inducing points. When the number of inducing points is fewer than 16, the upward trend in the map quality Q is conspicuous. The map quality progresses consistently with the increasing number of inducing points utilized in the training of the DGP model. As is shown in Table 3.1, the best map quality is achieved when the number of inducing points raise to 15. However, if the number of inducing points exceeds 16, the map quality drops considerably. Therefore, the number of inducing points is set as 15 for the implementation of the MapLoc system in the Broun Hall scenario.

3.6 Conclusions

In this paper, we proposed MapLoc, a bi-modal indoor localization system, to improve the location estimation accuracy. First, the DGP is used to regress uncertainty maps describing the signal distribution in the surveillance area. The artificial signal measurements that represent their own reliability are generated by sampling the signal distribution described by the mean and variance in the uncertainty map. In the artificial data generation, geometry constraints and user motion patterns are also taken into account. We then present a location prediction model to distinguish the effective signal measurements from the weak and fluctuating signals by learning the artificial signal measurements. The location prediction model leverages a stacked LSTM network as the backend. The auxiliary output is utilized to push the model to learn the signal map in the supervised training. The experimental results demonstrate that the location prediction model is able to choose the optimal signals among WiFi RSSI readings and geomagnetic measurements intelligently. Benefiting from the novel data generation method and location prediction model, the median error of the location estimation in both the public dataset and the Brun Hall dataset reach in centimeter-level accuracy.

Chapter 4

MulTLoc: A Framework for Multiple RFID Tag localization Using RF Hologram Tensors with Deep Neural Networks

4.1 Introduction

Radio-frequency identification (RFID) is an automatic identification technology that can read RFID tag data even when it is not in line of sight (LOS). It has been widely used in a variety of applications, including supply chain management, inventory tracking, access control, toll collection, and animal management. Due to its widespread use and low-cost tags, RFID technology has recently been expanded to fields such as healthcare monitoring and environmental sensing, owing to the rapid development of the Internet of Things (IoT). By exploiting the measurements in RFID readings, a rising variety of functions and applications are being added to existing RFID systems. For example, the systems for localization [17], gesture recognition [96], vital sign monitoring [97,98], pose estimation [99,100], temperature sensing [101], and material recognition [102], have attracted great interest from both industry and academia.

Among these existing and emerging applications, indoor localization has remained a hot research topic over the years, as it plays a critical role in solving position-related problems such as gesture recognition and human pose estimation. The RFID-based localization system is primarily based on two RFID measurements: the Received Signal Strength Indicator (RSSI) and the phase angle. SpotOn [103] used RSSI along with a path loss model to perform trilateration for indoor localization. LANDMARC [104] leveraged RSSI readings from reference tags as fingerprints to estimate an unknown tag position via fingerprint matching. The RFID phase angle is extremely sensitive to environmental changes as well, particularly variations in tag-antenna distance. Recent applications have achieved centimeter-level localization by predicting the direction of arrival (DoA) with the received RFID phase angle. SparseTag [17] used a spatial smoothing based method with a novel sparse RFID tag array to predict angles. RF-Wear [105] achieved a mean inaccuracy of 8-12° in tracking angles with a uniform linear array. Moreover, RF-Kinect [106] added a body geometry model to the RF hologram to determine limb orientation and human joint location.

On the other hand, deep neural networks have sparked a lot of interest and promise in domains like computer vision(CV) and natural language processing(NLP). To take advantage of the superior classification performance of deep networks, researchers integrate deep networks into indoor localization systems that collaborate with the fingerprinting method. Deep autoencoders, for example, were used to extract WiFi CSI features as fingerprints of the localization systems [28, 41, 47, 107]. With a deep residual sharing learning approach, ResLoc [108] enhanced localization accuracy. CiFi [109] was the first work to leverage a deep convolutional neural network (DCNN) for indoor localization. The generated AoA image was utilized for training a 6-layer DCNN.

Although the performance of such indoor localization systems improves with the iteration of deep networks, numerous intrinsic difficulties of fingerprinting-based localization systems remain unresolved. First, the minimum error of the fingerprinting-based localization system is determined by the distance between the stored fingerprints. To reduce the inherent inaccuracy, the number of fingerprints should be as large as possible. Apparently, it would be laborious or even impossible for some cases. Second, the fingerprints utilized in the system are highly linked to the equipment configuration. The AoA pictures utilized in the CiFi [109], for example, are defined by the configuration and setup of the receivers. Once the network has been trained, the receivers must be static. In other words, the network must be trained from scratch when a different setup or equipment is deployed. As a result, the transferability of CiFi is nearly zero. In this chapter, we try to decouple data creation from the hardware setup and to find a deep network that can accept inputs from various RFID device configurations. The tag position would be estimated with the deep network using inputs collected from any type of devices.

Therefore, we propose MulTLoc, a framework for Multiple RFID Tag localization utilizing RF Hologram tensors with deep neural networks, to alleviate the fundamental difficulties of the fingerprinting based method and to take advantage of deep neural networks. Radio frequency (RF) hologram tensors are created using phase readings from antenna pairs in the proposed framework. To generate ground truth tensors for supervised learning, a computer vision sensor (e.g., a Kinect V2) is used. Based on the DCNN and Swin Transformer [110], two representative hologram filter networks are investigated with the suggested framework to clean noisy input hologram tensors using the spatial relationship between tags. An intuitive peak detection technique will be used to infer the location of RFID tags.

The main contributions made in this chapter are summarized as follows.

- To the best of our knowledge, this is the first study to utilize RF hologram tensors to train deep networks for three-dimensional localization. The use of the RF hologram tensor renders deep networks independent of environmental changes, considerably improving the robustness and transferability of the proposed system.

- We implemented two novel deep networks to clean up RF hologram tensors. In the networks, the spatial information between multiple tags is leveraged to suppress the fake peaks that exist in the original RF hologram tensors. We begin by introducing a DCNN-based network for cleaning RF hologram tensors. A Swin Transformer based network is also proposed to filter RF hologram tensors. In the Swin Transformer training, self-supervised learning is utilized to extract general features from hologram tensors. Position estimation is reduced to a simple peak detection problem that can be performed fast with the sanitized hologram tensor.

- A prototype of the proposed MulTLoc framework is built using the commercial off-the-shelf (COTS) RFID devices. With a multiple-joint localization experiment, the performance of the proposed framework is evaluated. The experimental results show that the MulTLoc framework is capable of simultaneously localizing multiple tags in three dimensions.

The remainder of this chapter is organized as follows. We present an overview of related work in Section 4.2. Section 4.3 introduces the preliminaries and motivation of our approach. We present the MulTLoc design in Section 4.4 and our experimental study in Section 4.5. Then, Section 4.6 concludes this chapter.

4.2 Related Work

Indoor localization is crucial in RF sensing for resolving position-related concerns. With the development of mobile communication technology over the last decade, academia and industry have paid close attention to location-based services. Signal processing has long been used to determine the position of a signal source by estimating the Time-of-Flight, Angle-of-Arrival, or a third signal parameter such as doppler shift and Angle-of-Departure [17, 23, 82, 111–113]. The accuracy of parameter estimate, however, is governed by the number of antennas (for AoA) and the transmission frequency bandwidth (for ToF), which are often fixed in a certain wireless communication system. As a result, the expense of improving parameter estimate would be prohibitively expensive.

On the other hand, the fingerprinting method emerges with its convenience and effectiveness, which transfers the localization problem into a feature matching to estimate the location of the signal. Researchers are working on two tracks to increase the accuracy of fingerprint-based localization. First, more and more powerful classification algorithms are introduced to the fingerprinting-based localization. K-nearest neighbors algorithm (KNN) and its modification are commonly leveraged in the indoor localization system [25, 114–116]. The machine learning algorithms, such as Random forest [117, 118] and AdaBoost [119, 120], are often used in promoting the performance of the classification as well. Another important aspect influencing the localization accuracy of the fingerprinting-based localization system is the quality of the fingerprints. Principal component analysis (PCA) is a common tool to extract features from the original fingerprints [121, 122] for enhancing the fingerprint quality. Recently, with the development of deep learning, deep autoencoder has been implemented in the fingerprinting based localization systems as feature extractors [29, 41]. Since deep networks show the superior performance in the image classification task, the feature extraction and classification are unified in the fingerprinting based localization systems using deep neural networks. The features from the AoA images are extracted and classified in CiFi [109] and ResLoc [123] with one effort. Despite the fact that new techniques are always being developed to improve the performance of fingerprinting-based localization systems, the inherent flaws of the fingerprinting method are not avoided, where the localization accuracy is determined by the density of the fingerprints, and any changes in fingerprints would trigger the update of the system. In this research, we will attempt to present a novel framework for avoiding these fingerprint-related issues.

Deep neural network, as previously stated, has been frequently used in indoor localization systems because of the excellent feature extraction and classification capabilities. It has evolved over the last decade to meet the needs of various downstream jobs. Since the debut of AlexNet [124], DCNN has become a superstar in computer vision. ResNet [125] constructs a DCNN with hundreds of layers by utilizing shortcut connections. Hourglasss [126] and U-Net [127] use an encoder-decoder design to keep the high-resolution representation of the images. To this day, CNN remains the key backbone for addressing computer vision challenges. In the area of NLP, recurrent neural networks (RNN) were prominent for dealing with temporal sequences [128, 129]. Transformer [130] has recently emerged as a dominating successor by using an attention method to construct global interdependence between input and output. The transformer is also applied for the computer vision tasks. Vision Transformer(ViT), Swin transformer, and their modifications [110, 131, 132] keep improving the state-of-the-art performance in various CV tasks as backbones. Based on the proposed framework, we deployed two representative networks, DCNN and swin transfomer, as the backbone for implementing the hologram filter network.

4.3 Preliminaries and Motivation

4.3.1 RFID Phase Model

Sensitive and trustworthy measures should be taken from the original RFID readings in order to locate RFID tags in real-time. In contrast to RSSI, the phase value is commonly used in many RFID-based sensing applications [101, 133, 134]. As shown in (4.1), the phase reading $\theta_{i,m}$ is a periodic function with a period of 2π.

$$\theta_{i,m} = \mod\left(\frac{4\pi\,|TA_m|}{\lambda_i} + \theta_{tag} + \theta_{equipment}, 2\pi\right), \tag{4.1}$$

where $|TA_m|$ denotes the distance between the tag T and the antenna A_m; λ_i is determined by the frequency of channel i; θ_{tag} and $\theta_{equipment}$ are the phase offsets caused by the RFID tag and RFID hardware such as antenna and reader, respectively. $\theta_{equipment}$ is a constant for a given RFID system; hence it could be removed conveniently.

4.3.2 Hologram Tensor

Tagoram [19] is the first to introduce the concept of an RF hologram for RFID indoor localization. The primary concept underlying an RF hologram is to compute the similarities between theoretical and measured phase values for each grid in the surveillance space. To eliminate the tag-related phase offset, i.e., θ_{tag} in (4.1), we use the phase difference as the observation in our system. The real phase difference obtained with the phases collected from an antenna pair (m, n) on channel i is denoted as

$$p_{i,m,n} = \mod(\theta_{i,m} - \theta_{i,n}, 2\pi). \tag{4.2}$$

When the coordinates of the two antennas are known, the theoretical phase difference between antenna pair (m, n) on channel i can be determined. The theoretical phase difference at the grid position $G_{x,y,z}$ for the antenna pair (m, n) is shown as

$$q_{i,m,n}^{x,y,z} = \mod\left(\frac{4\pi\,|G_{x,y,z}A_m|}{\lambda_i} - \frac{4\pi\,|G_{x,y,z}A_n|}{\lambda_i}, 2\pi\right). \tag{4.3}$$

With the real and theoretical phase differences, their similarity, $S_{x,y,z}$, is estimated as follows.

$$\begin{aligned} S_{x,y,z} &= \sum_{(M,N)} \sum_{I} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\delta_{i,m,n}^{x,y,z})^2}{2\sigma^2}\right) \\ \delta_{i,m,n}^{x,y,z} &= \ \mathrm{mod}\left(p_{i,m,n} - q_{i,m,n}^{x,y,z}, 2\pi\right), \end{aligned} \tag{4.4}$$

where (M, N) represents the set consisting of all available antenna pairs, I denotes the set of all available channel indices. The hologram tensor, $\mathbf{S}$, is constructed as

$$\mathbf{S} = \begin{bmatrix} S_{1,1,z} & S_{1,2,z} & \cdots & S_{1,y,z} \\ S_{2,1,z} & S_{2,2,z} & \cdots & S_{2,y,z} \\ \vdots & \vdots & \ddots & \vdots \\ S_{x,1,z} & S_{x,2,z} & \cdots & S_{x,y,z} \end{bmatrix}, z = 1, 2, ..., Z, \tag{4.5}$$

where each element is scaled to have a value in $[0, 1]$ in the proposed system.

4.3.3 Motivation

MulTLoc is, to the best of our knowledge, the first effort to train deep learning models for real-time three-dimensional localization using hologram tensors. Although some indoor localization systems, e.g., [61, 109, 123], use radio frequency signals to produce images or tensors for offline training, the generated data may lack a strong relationship between the observation and the spatial location. In these applications, images and tensors are employed as fingerprints, and deep networks are used as classifiers. The ambiguity between fingerprints may be lost throughout the dataset construction process, restricting the transferability of the localization model. In comparison to the images and tensors in the preceding studies, the hologram tensor is interpretable. The hologram tensors represent the possibility of a tag being located at a grid position in the surveillance space. The similarity S is directly connected to the distances between the tag and the antennas, and it is highly independent of the equipment used to generate the tensor.

A hologram matrix formed in a two-dimensional area is shown in Fig. 4.1. It displays the two-dimensional projection of the hologram matrix. The exact location of the target tag is indicated by the red pentagram. As can be seen, there is a peak near the position of the ground truth. However, because of the multipath and phase wrapping effects, multiple fake peaks are formed and distributed across the hologram. Some of the phony peaks have even greater similarity values. To avoid such issues, data prepossessing has become a key component of many RFID-based sensing systems. Some ways improve accuracy at the expense of real-time performance. Channel selection [17] and phase sanitation [135], for example, are used to keep systems away from phase readings tainted by the multipath effect. Such approaches, however, may be impractical for real-time localization systems. This is because multiple-round interrogations are required, and the tag (or target) will not remain stationary until the system performs a sufficient number of interrogations. Moreover, some applications rely on specific hardware and deployment, such as the synthetic-aperture array [136] and multi-resolution filtering [137], to mitigate the detrimental effect caused by the phase wrapping ambiguity. Despite the fact that these technologies offer sufficient precision and real-time performance, the need for customized hardware raises costs and limits the compatibility with COTS RFID systems. Furthermore, tag localization in three dimensions is a more difficult challenge than in two dimensions. In this chapter, we presents two unique neural network with the proposed framework to handle such issues.

4.4 Overview of the MulTLoc System

In this chapter, we present MulTLoc, an RFID-based localization framework for estimating the location of multiple tags *simultaneously* utilizing noisy hologram tensors. Despite the fact that MulTLoc, like most previous deep learning-based localization systems, is trained with ground truths provided by sensors such as an RGB-D camera, the localization problem is treated as *regression* in this study. To estimate the coordinates of unknown sites, traditional fingerprinting methods leverage deep neural networks to treat location estimation as a *classification* problem. The size of the fingerprint database limits the accuracy of localization, and the granularity of the

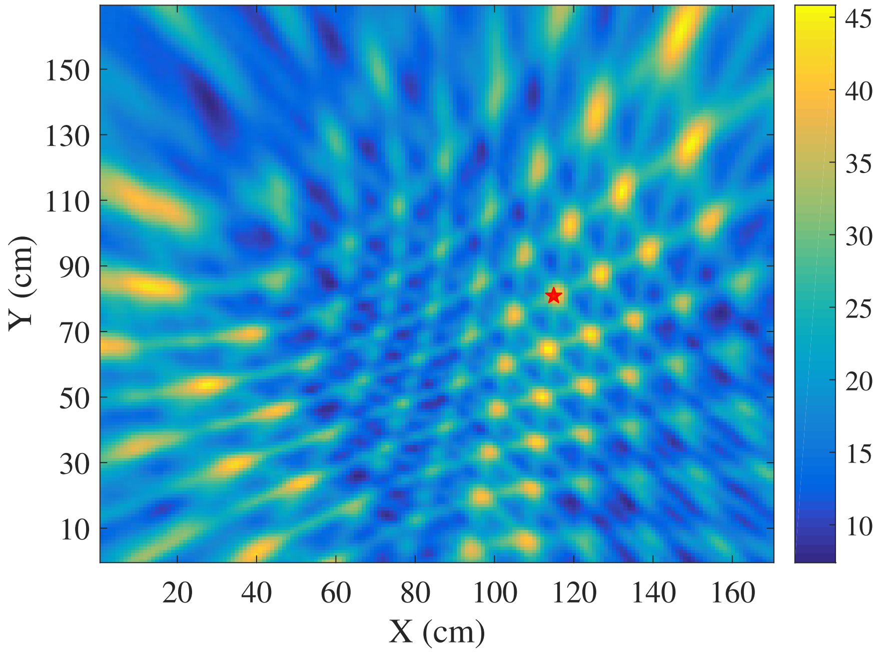


Figure 4.1: Hologram of a 2D scenario. The red pentagram denotes the ground truth.

fingerprints determines the inherent inaccuracy of the system. The network would not give location estimation instantaneously in the MulTLoc framework. Instead, noisy hologram tensors are regressed to single-peak hologram tensors, which are free from the fake peaks created by the multipath and phase wrapping effects. A location estimate might be performed intuitively using the sanitized hologram tensor.

4.4.1 MulTLoc System Architecture

Fig. 4.2 depicts the MulTLoc architecture. An RFID system collaborates with a vision-based sensor to generate the hologram tensors and the accompanying ground truth tensors for training the hologram filtering networks. Because the hologram tensors and ground truth coordinates provided by the vision-based sensor are typically in distinct coordinate systems, our proposed framework uses the Robot Operating System (ROS) to synchronize and unify the data acquired from diverse hardware. Generally, any deep neural network that is capable of sanitizing noisy hologram tensors would be compatible with the MulTLoc. We utilized two typical neural networks in this chapter to evaluate the performance of the proposed framework. The final position estimation would be induced conveniently using a simple peak detection algorithm with the sanitized hologram tensors.

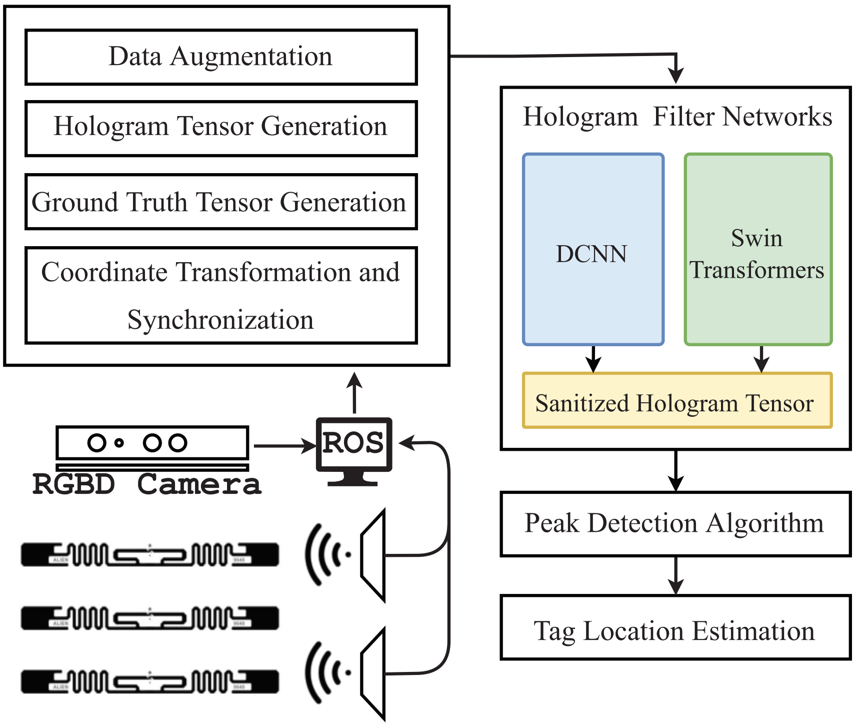

Data Augmentation
Hologram Tensor Generation
Ground Truth Tensor Generation
Coordinate Transformation and Synchronization
RGBD Camera
ROS
Hologram Filter Networks
DCNN
Swin Transformers
Sanitized Hologram Tensor
Peak Detection Algorithm
Tag Location Estimation

Figure 4.2: The MulTLoc system architecture.

Based on the proposed framework, we deploy two representative deep network backbones for creating hologram filter networks. First, a DCNN-based hologram filter network is designed with the hourglass backbone to clean and compress the noisy hologram tensors. To recover the original size of the hologram tensor, trilinear interpolation or equivalent approaches would be used before the peak detection. Data augmentation is used to prevent overfitting in the training of the DCNN-based hologram filter network. However, the DCNN network architecture in this chapter is related to the channel of the input tensors. In Fig. 4.3, three residual units are used to sanitize the hologram tensors from three tags, however additional residual units are required to cope with more channels. To resolve the issue, another hologram filter network is also proposed with the Swin Transformer for keeping architecture stable in taking care the tensors from more tags. The output of the network keeps the original size of the input tensors, which would be directly adopted for location prediction. Self-supervised learning is deployed in the training for extracting latent features from noisy hologram tensors. Once the networks have been properly trained, the vision-based sensor will no longer be required for location estimation.

4.4.2 Training Dataset Generation

The hologram tensor must be labeled with the relevant ground truth tensor in order to train the networks successfully. However, the ground truth coordinates and hologram tensors are acquired by different sensors with distinct coordinate systems. The reported coordinates for most vision-based sensors are normally determined by the coordinate origin of the sensor space. For example, the center of the depth sensor is the origin of the coordinates for Kinect V2, whereas the surveillance space determines the coordinates of the antennae in MulTLoc. ROS is used in MulTLoc to integrate the hologram tensors from the RFID system and the coordinates from the vision-based sensor to label hologram tensors with accurate ground truth tensors. We transfer all coordinates from the vision-based sensor into the frames of the hologram tensors depending on the sensor pose and position in the surveillance space. Meanwhile, for synchronization, timestamps are appended to both the hologram tensors and the ground truth coordinates. An RF hologram tensor will be assigned to the coordinates with the most recent timestamp.

The ground truth tensor, $\mathbf{K}$, is constructed using a gaussian kernel. Based on the synchronized ground truth coordinates by measuring the Euclidean distance $|G_{x,y,z}H|$ between the gird location $G_{x,y,z}$ and the ground truth location H, the ground truth tensor, $\mathbf{K}$, is formulated as

$$\mathbf{K} = \begin{bmatrix} K_{1,1,z} & K_{1,2,z} & \cdots & K_{1,y,z} \\ K_{2,1,z} & K_{2,2,z} & \cdots & K_{2,y,z} \\ \vdots & \vdots & \ddots & \vdots \\ K_{x,1,z} & K_{x,2,z} & \cdots & K_{x,y,z} \end{bmatrix}, z = 1, 2, ..., Z, \tag{4.6}$$

where each element of $\mathbf{K}$ is given by

$$K_{x,y,z} = \frac{1}{\epsilon\sqrt{2\pi}} \exp\left(-\frac{|G_{x,y,z}H|^2}{2\epsilon^2}\right). \tag{4.7}$$

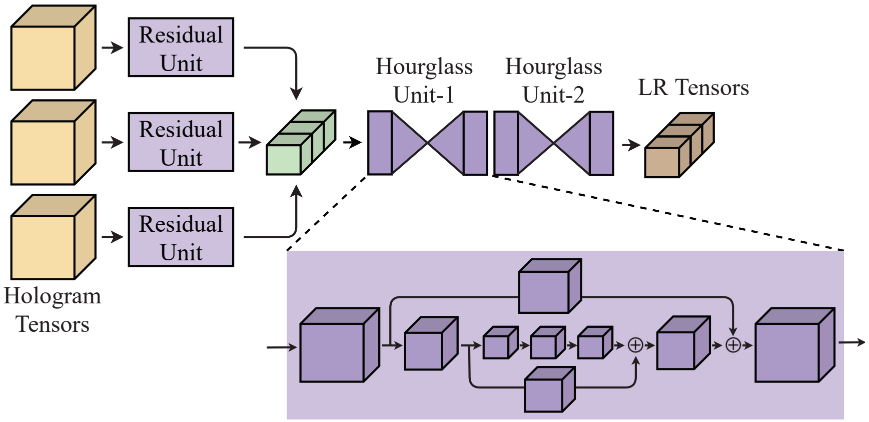


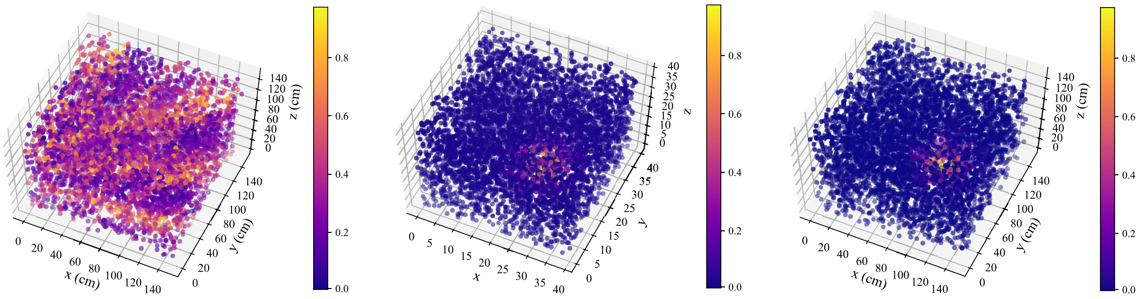
Figure 4.3: Architecture of the DCNN based hologram filter network.

where ϵ controls the radius of the ground truth peak. In the MulTLoc framework, $\mathbf{K}$ supervises the training of hologram filter networks. To coordinate the compressed output of the DCNN-based hologram filter network, the ground truth tensor $\mathbf{K}$ is downsampled. Since the hologram tensor is interpretable spatially, our training dataset is augmented by the flipping and rotating operations in the training of the DCNN based hologram filter network.

4.4.3 Design of DCNN for Filtering Hologram Tensors

As shown in Fig. 4.3, a DCNN-based hologram filter network is introduced to remove the fake peaks from the hologram tensors. In contrast to the changing settings that compromise the efficacy of fingerprinting-based localization systems, the positional connection between tags is relatively constant, particularly for passive tags attached to items. The hologram filter network is intended to learn the spatial connection between tags in order to differentiate the real peaks in the RF hologram tensors. We downsample the hologram tensors from n tags and concatenate them into an n-channel tensor using residual units to reduce the amount of weights in the proposed network and accelerate training. The newly created n-channel tensor retains the detailed information in the original hologram tensors while also including a coherent understanding among the tags. In our following experiments, n is set to three to locate three tags at the same time.

The residual unit of the hologram filter network consists of two residual blocks [9]. In each block, two three-dimensional convolutional layers are included. The hourglass blocks [126]



(a) (b) (c)

Figure 4.4: (a) The original hologram tensors. (b) The filtered hologram tensor. (c) The full-size ground truth tensor.

are arranged end-to-end following the residual blocks as the backbone of the hologram filter network to extract features in the n-channel tensor at different sizes. The design of the hourglass unit is comparable to that of an encoder-decoder network, as seen in Fig. 4.3. The input tensor is first compressed and then upscaled in the unit. Each purple cube in the hourglass unit is made up of three residual blocks, each of which has three three-dimensional convolutional layers. To maintain spatial information at different resolutions, the skip connection is used between blocks of the same size. The bottom-up, top-down inference is repeated by stacking the hourglass units. By computing the loss between the ground truth tensors and the output tensors, the deep network is optimized with the Adam algorithm. We would discuss the selection of loss function in the following section. For accelerating training, intermediate supervision is applied at each hourglass unit in the DCNN-based hologram filter network. The hologram filter network produces a low resolution, n-channel tensor (i.e., the LR Tensor), which is divided into n low resolution hologram tensors for location estimation.

Fig. 4.4(a) and Fig. 4.4(b) display the input and output of the hologram filter network, respectively. Lower similarity values are shown as bluish pixels in the figures. The RF hologram tensor is produced using the phases gathered from our testbed, which covers a region of dimension $1.5m \times 1.5m \times 1.5m$ (see Section 4.5.1 for details). The fake peaks spread out in the hologram tensor, similar to the hologram matrix in the two-dimensional case. Despite the fact that the space has four bands with greater similarity values, no clear peak can be recognized. The hologram filter network generates the sanitized hologram tensor, shown in Fig. 4.4(b), by mixing the holograms from three tags. The majority of the fake peaks in the input tensor has

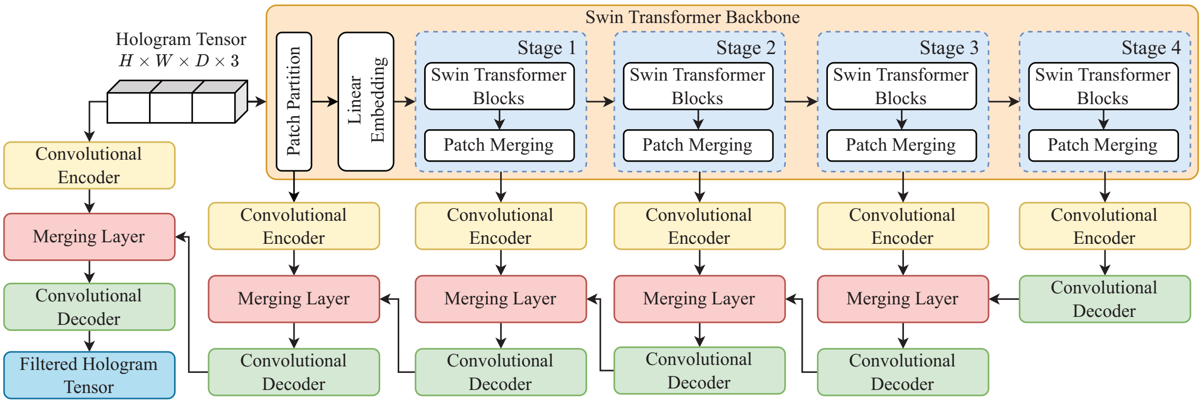


Figure 4.5: Architecture of the Swin transformer based hologram filter network.

now been muted. The single bright spot is in the center of the filtered hologram tensor, which is similar to the ground truth tensor in Fig. 4.4(c). The spatial connection among the tags has now retrieved the location information concealed under fake peaks.

4.4.4 Design of Swin Transformers for Filtering Hologram Tensors

However, the cleaned tensor is compressed by the DCNN based network in Fig. 4.4(b). Because the input of our networks is a 4D tensor, for example it consisting of $150 \times 150 \times 150 \times 3$ pixels in the following experiment, and the number of parameters escalates with the use of 3D convolution, the output size is reduced to save memory while utilizing the DCNN as the backbone. Data compression appears to be at the expense of location estimation accuracy. Furthermore, the number of input channels determines the architecture of the DCNN network. For dealing with the tensors from three tags, three residual units are included. More residual units must be added to the network if the network is used to localize more tags. Thus, the framework's compatibility is still limited by the DCNN backbone.

To address the issue, a Swin Transformer based network is used to sanitize the noisy hologram tensors. In our framework, it is equivalent to the DCNN-based network. The Swin Transformer backbone is not only robust to the different size of the input tensor but also has the output with same size as the input tensor. The architecture of the Swin Transformer based hologram filter network is depicted in Fig. 4.5. The network is a 3D variation of the U-Net [127] with a Swin Transformer backend. The input tensor is first split into non-overlapping 3D tokens

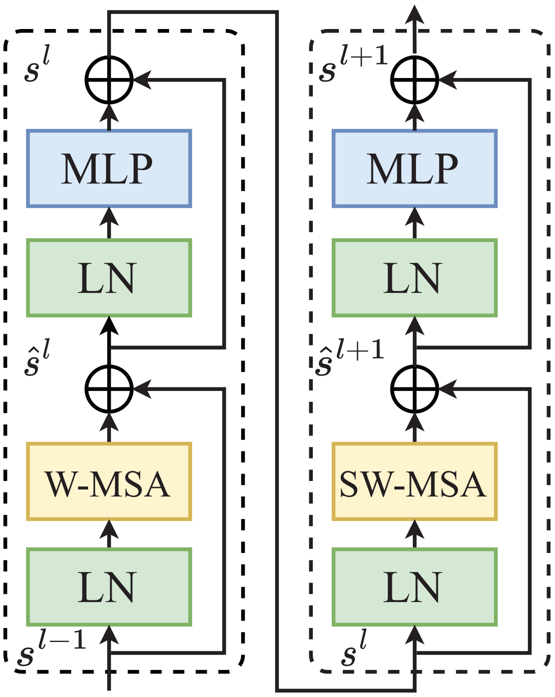


Figure 4.6: Swin transformer blocks.

for feeding into the Swin Transformer blocks. Our implementation sets the patch size to 2×2×2. It consists of a raw feature dimension of $2 \times 2 \times 2 \times 3$ associated with the multi-tag hologram tensor with 3 channels. The raw features are projected into a 48-dimensional space using a linear embedding layer, which is consistent with the traditional Swin Transformer. Then, the processed tokens are applied with Swin Transformer blocks. Fig. 4.6 exhibits the shifted window based self-attention in the Swin Transformer blocks, where W-MSA and SW-MSA represent the regular window based multi-head self-attention (MSA) and shifted window based MSA, respectively. A LayerNorm(LN) layer is adopted before each MSA and MLP. The tokens are first partitioned into small cubes in the Swin Transformer block. For example, the token of H×W×D would be divided in to $\frac{H}{M}$×$\frac{W}{M}$×$\frac{D}{M}$ cubes with a window of M×M×M. In the following block, the window would shift $(\frac{M}{2}, \frac{M}{2}, \frac{M}{2})$ pixels, so that the connection between neighboring non-overlapping windows in the previous block would be introduced in the network. With the approach, the output of two consecutive Swin Transformer blocks is computed as,

$$
\begin{aligned}
\hat{s}^l &= \text{W-MSA}(\text{LN}(s^{l-1})) + s^{l-1} \\
s^l &= \text{MLP}(\text{LN}(\hat{s}^l)) + \hat{s}^l \\
\hat{s}^{l+1} &= \text{SW-MSA}(\text{LN}(s^l)) + s^l \\
s^{l+1} &= \text{MLP}(\text{LN}(\hat{s}^{l+1})) + \hat{s}^{l+1} \qquad (4.8)
\end{aligned}
$$

The self-attention is given as

$$
\text{Attention}(Q, K, V) = \text{SoftMax}(\frac{QK^T}{\sqrt{d}} + B)V \qquad (4.9)
$$

where Q, K, V stand for queries, keys, and values, respectively. d is the scale-down factor, and B is the relative position bias. A patch merging layer always follows the Swin Transformer blocks for shrinking the size of the features by a factor of 2 in each stage. The output of each stage would not only be passed on to the subsequent stage, but also be fed to DCNN-based decoders to regenerate the filtered hologram tensor.

The feature representation from the Swin Transformer backbone is first adjusted with a convolutional encoder before it is concatenated with the features from the decoder of the lower layer. The convolutional decoder processes the merged features, and the output returns to the higher layer. In our implementation, the filtered hologram tensor is obtained directly with the convolutional decoder from the top layer. To supervise the training of the network, a mixed loss function is formulated as,

$$
L_{mix} = \alpha L_{\text{MS-SSIM}} + (1 - \alpha) \times L_{\ell_1} \qquad (4.10)
$$

where $L_{\text{MS-SSIM}}$ is the multiscale structural similarity index, L_{ℓ_1} represent ℓ_1 loss, α is a hyper-parameter [138].

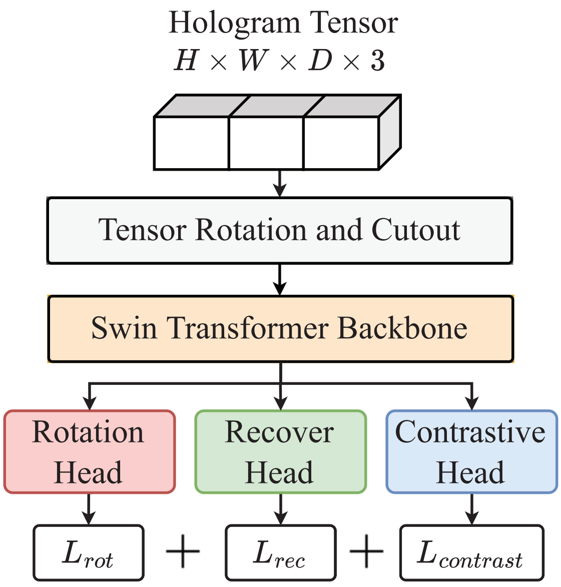
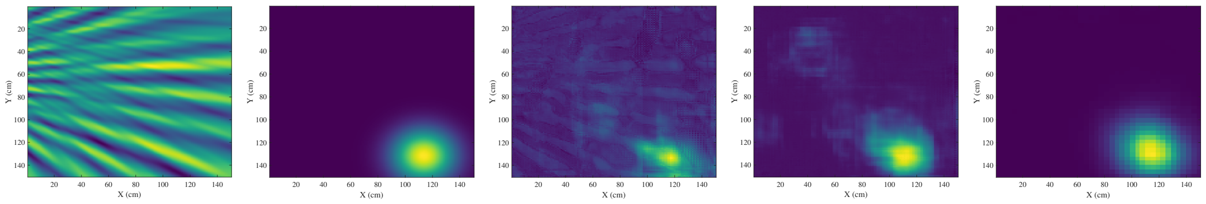


Figure 4.7: Architecture of self-supervised pre-training.



(a) (b) (c) (d) (e)

Figure 4.8: (a) A slice of the input tensor. (b) A slice of the ground truth tensor. (c) A slice of the sanitized tensor using self-supervised pre-train. (d) A slice of the sanitized tensor using supervised learning.(e) A slice of the sanitized tensor obtained with DCNN-based network

4.4.5 Self-supervised Pre-training of Swin Transformers

Self-supervised pre-training has made a significant contribution to the development of cutting-edge models for a wide range of NLP tasks. Recent research has revealed numerous self-supervised ways for enhancing the capacity of deep neural networks to learn feature representations in vision tasks as well [139, 140]. In this chapter, the self-supervised pre-training is leveraged with the Swin Transformer based hologram filter network to promote the performance of sanitizing noisy tensors in the proposed framework.

According to Fig. 4.7, we adopted three pretext loss for learning a good data representation in the self-supervised pre-training, which are inspired by the prior work for medical image analysis [141]. The input hologram tensor **S** is first cropped and rotated to generate sub-volumes randomly. Using the Swin Transformer backbone, the feature representation is extracted from the sub-volumes, and three different projection heads are leveraged to achieve the corresponding pretext tasks. The first task is to predict the angle rotation of the sub-volumes in 6 classes including $90°$, $-90°$ along with x-axis, y-axis, and z-axis respectively. The cross-entropy loss is utilized as below

$$L_{rot} = -\sum_{m=1}^{6} r_m \text{log}(\hat{r}_m) \tag{4.11}$$

where $\hat{r}_m$ is the softmax result from the rotation head, r_m is the ground truth.

Tensor recover task is also a part of the self-supervised pre-training. We mask out a portion of pixels in the sub-volumes with a ratio s. The sub-pixel convolution in the recover head regenerates the masked pixels with the feature representation from the Swin Transformer backbone. The MSE loss L_{rec} is leveraged to measure the difference between the ground truth sub-volume S_{sub} and the recovered sub-volume $\hat{S}_{sub}$, which is given as,

$$L_{rec} = \frac{1}{P}\sum(S_{sub} - \hat{S}_{sub})^2 \tag{4.12}$$

where P is the number of pixels in the sub-volume.

Contrastive learning [139] is also a part of the self-supervised training in the proposed framework. We leverage a simple instance discrimination task as the pretext task. Two correlated sub-volumes of the input hologram tensor, $\mathbf{s}$ and $\mathbf{s}_+$, are generated with the tensor rotation and cutout at first. With the Swin Transformer backbone and the contrastive head, the feature representations are extracted from $\mathbf{s}$ and $\mathbf{s}_+$ and denoted as q and k_0. For a minibatch of N tensors, only the feature representation from the same input tensor is treated as positive pair, while the feature representation $\{k_1, k_2, \cdots k_{N-1}\}$ from the rest $N-1$ tensors are the negative examples. Apparently, the instance discrimination task is actually a N-way classification problem, which tries to class q as k_0. Thus, the contrastive loss function, called InfoNCE [142], is defined as

$$L_{contrast} = -\log \frac{\exp(q \cdot k_0/\alpha)}{\sum_{m=0}^{N-1} \exp(q \cdot k_m/\alpha)} \tag{4.13}$$

where the dot product is implemented to measure the similarity between the feature representation, α is a temperature hyper-parameter.

Finally, the Swin Transformer backbone is self-supervised by minimizing the complex loss function as follow:

$$L = L_{rot} + L_{rec} + L_{contrast} \tag{4.14}$$

Our Swin Transformer based hologram filter network is fine-tuned using regular supervised learning after self-supervised pre-training. Fig.4.8 depicts the progress brought by self-supervised pre-training. Fig.4.8(a) and Fig.4.8(b) show a noisy hologram tensor slice and the corresponding ground truth slice. In Fig.4.8(c), a sanitized slice of the hologram tensor is generated using the pre-trained weight, while Fig.4.8(d) shows the slice sanitized by the network without the self-supervised pre-training. By comparing Fig.4.8(a) and Fig.4.8(c), we notice that the band pattern in the Fig.4.8(a) is extracted and recovered in Fig.4.8(c). The peak spot locates at one of the bands in the slice, which meets our observation in Fig.4.1. The band pattern, however, vanishes in Fig.4.8(d). Instead, the shadow area in Fig.4.8(d) is consistent with the blur area in Fig.4.8(a). It appears that the network learns how to sanitize the tensor via a ”shortcut”, which is not our expectation. Furthermore, the area of the peak spot in Fig.4.8(c) is significantly denser than those in Fig.4.8(d). It reveals that self-supervised pre-training is able to improve location estimation by extracting detailed and interpretative feature representations from noisy tensor inputs. Furthermore, Fig.4.8(e) displays a slice of the sanitized tensor from the DCNN-based hologram filter network. Compared to the previous slices, Fig.4.8(e) is almost identical to the ground truth slice in Fig.4.8(b). Even though the slice is much cleaner than the slices from the Swin Transformers, the details from the original input tensors are eliminated through the DCNN-based network. It is difficult for us to discover how the network cleans up the tensor in the forward propagation. The phenomenon exhibits the difference between the DCNN backbone and the Swin Transformer backbone. The Swin Transformer has a larger representation capacity, e.g. the effective features are kept in Fig.4.8(c). However, it usually suffers from data shortage because of the lack of the typical convolutional inductive bias. On the other hand, the DCNN backbone performs well even with a dataset of limited size. The detailed comparison between the two backbones will be presented in the following chapter.

4.4.6 Location Estimation

The tag location could be inferred easily using a simple peak detection algorithm with the sanitized hologram tensors. Because the sanitized tensor from the DCNN-based hologram filter network is compressed, we employ trilinear interpolation to recover its size. The estimation location $\widehat{G}$ is computed as follows.

$$\widehat{G} = \{G | f(\mathbf{S}_R, G) = \max(\mathbf{S}_R)\}, \tag{4.15}$$

where $f(\cdot)$ extracts the similarity value at the grid location G from the sanitized hologram tensor $\mathbf{S}_R$.

4.5 Experimental Study

4.5.1 Testbed Configuration

To evaluate the performance of the proposed framework, we built a prototype using a Zebra FX9600 reader and eight Zebra AN720 antennas. Three UPM Raflatac Frog 3D tags are utilized as localization targets. In the experiment, we assess the performance of the proposed framework by concurrently localizing the tags affixed to the human body. A Kinect V2 device collaborates with a three-dimensional human position estimation algorithm [143] to produce ground truth coordinates for supervised learning. For dataset creation and tag position estimation, the target tags are mounted to the shoulders and neck. ROS Kinetic Kame is utilized to synchronize and unify the coordinates and tensors from the Kinect V2 and the RFID reader. We adjust the requirement for hologram tensor creation to ensure real-time performance of the proposed

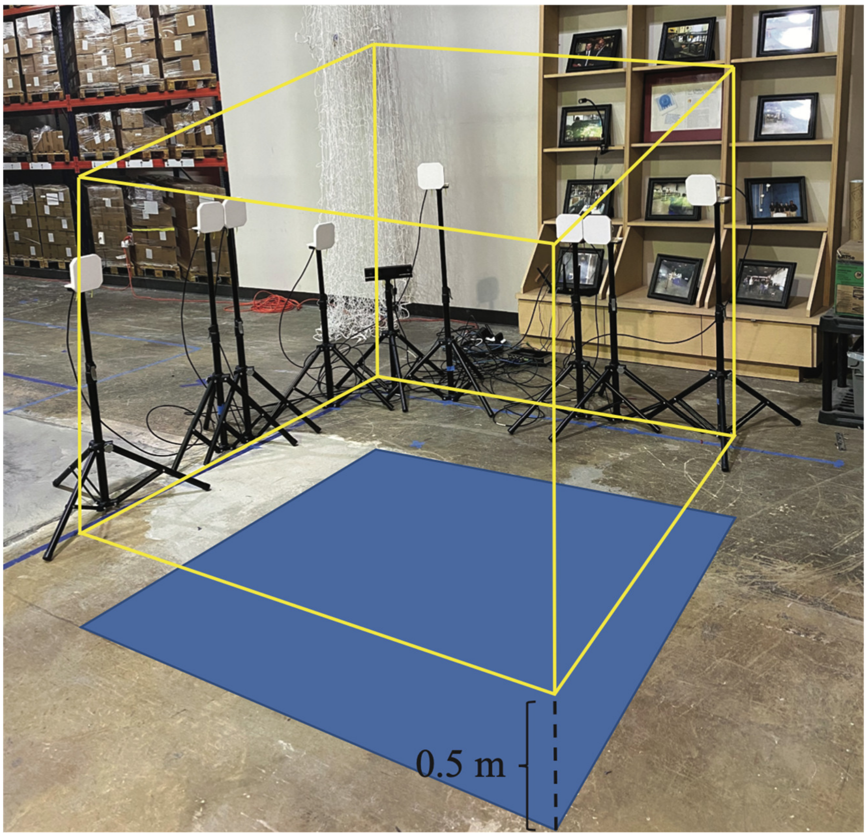


Figure 4.9: The MulTLoc testbed setup.

framework. When five antenna pairs are available, the phases from seven channels will be used to construct the hologram tensors. The yellow lines in Fig. 4.9 outline the surveillance space of the prototype, which covers a space of dimension $1.5m \times 1.5m \times 1.5m$ at $0.5m$ above the ground. The grid size is set at $1cm$. Furthermore, the similarities at each grid position in the surveillance space are computed in parallel using CUDA GPU programming to speed up the construction of hologram and ground truth tensors.

Table 4.1 and Table 4.2 illustrate the backbone detail of two hologram filter networks. For the DCNN-based hologram filter network, each residual unit includes two ResUnit_block. The downsampled tensors are concatenated as a multi-channel tensor (green cubes in Fig. 4.3) and then processed by a convolutional layer (conv2 in Table 4.1). The hourglass units are stacked end-to-end for filtering noisy hologram tensors. Multiple hourglass units could be leveraged in the DCNN-based hologram filter network. The number of hourglass units would not affect the utilization of Hourglass_input and Hourglass_output in the network. An extensive discussion about the effect of the number of hourglass units on the location estimation will be given in the

layer name	output size	kernel size
conv1	$(80 \times 80 \times 80)$	$\begin{bmatrix} 3 \times 3 \times 3 \end{bmatrix}$
ResUnit_block1	$(80 \times 80 \times 80)$	$\begin{bmatrix} 3 \times 3 \times 3 \end{bmatrix} \times 2$
ResUnit_block2	$(40 \times 40 \times 40)$	$\begin{bmatrix} 3 \times 3 \times 3 \end{bmatrix} \times 2$
concat	$(40 \times 40 \times 40)$	Null
conv2	$(40 \times 40 \times 40)$	$\begin{bmatrix} 3 \times 3 \times 3 \end{bmatrix}$
Hourglass_input	$(40 \times 40 \times 40)$	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \\ 1 \times 1 \times 1 \end{bmatrix}$
Hourglass	$(40 \times 40 \times 40)$	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \\ 1 \times 1 \times 1 \end{bmatrix} \times 4^*$
Hourglass_output	$(40 \times 40 \times 40)$	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \\ 1 \times 1 \times 1 \end{bmatrix} \times 4$

Table 4.1: The detail of DCNN-based hologram filter network. (Each purple cube, representing the feature tensor, is processed with the displayed kernel. Downsampling and upsampling layers are not shown in the table.)

	Stage-1	Stage-2	Stage-3	Stage-4
Layer Size	$\begin{bmatrix} \text{win. sz. } 7{\times}7{\times}7 \\ \text{dim } 48, \text{head } 3 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{win. sz. } 7{\times}7{\times}7 \\ \text{dim } 48{\times}2, \text{head } 6 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{win. sz. } 7{\times}7{\times}7 \\ \text{dim } 48{\times}4, \text{head } 12 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{win. sz. } 7{\times}7{\times}7 \\ \text{dim } 48{\times}8, \text{head } 24 \end{bmatrix} \times 2$
Output Size	$(24{\times}24{\times}24)$	$(12{\times}12{\times}12)$	$(6{\times}6{\times}6)$	$(3{\times}3{\times}3)$

Table 4.2: The detail of Swin Transformer backbone.

rest of the paper. The detailed information about the Swin Transformer backbone is presented in Table 4.2. As is shown in Fig. 4.5, four stages are included in the Swin Transformer backbone. The feature dimension and number of attention heads would increase by a factor of two in each stage. With the patch merging, the output size from stages is shrunk by a factor of two. Two blocks are included in each stage by default. We will discuss the effect of feature dimension and the number of blocks on the performance of location estimation in the following paragraphs.

To train the deep networks in the MulTLoc framework, we collect tensors and related coordinates from several volunteers who are attached with three tags and move randomly in the surveillance space. Three hundred groups of data are included in the dataset. In each group, two tensors are from the shoulder tags and one tensor is from the neck tag. The acquired data divided randomly for training, validation, and testing. 80% percent of the tensor groups are

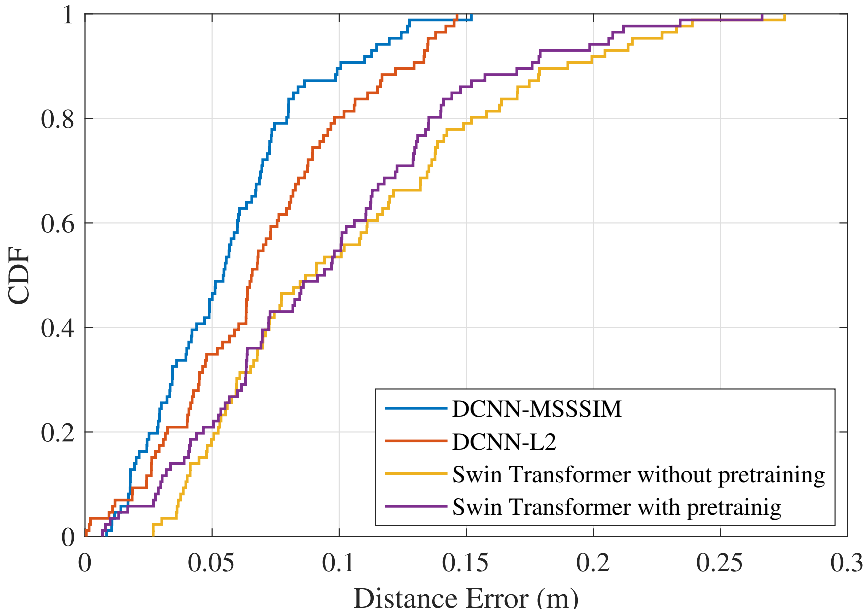


Figure 4.10: CDFs of location estimation errors with different hologram filter networks

used to train deep neural networks. The training dataset includes seven hundred and twenty RF hologram tensors in total, while the remaining sixty tensor groups are evenly sperated for validation and testing. Furthermore, an Nvidia RTX3090 GPU and an RTX A6000 GPU are utilized to accelerate the computation of the two deep networks.

4.5.2 Experiment Results and Discussions

Fig. 4.10 presents the cumulative distribution function (CDF) of localization errors, which exhibits the overall localization precision brought by different network configurations. The best localization performance is achieved by the DCNN backbone cooperating with the MSSSIM loss function, which has a mean error of $0.0558m$. When the L2 loss is leveraged in the DCNN training, the mean localization error increases to $0.0688m$. For the Swin Transformer based network, a mean error of $0.0961m$ is achieved when self-supervised learning is leveraged in training, whereas the mean error is $0.1041m$ without the self-supervised training. Even though a precision improvement in location estimation is brought by the self-supervised training, DCNN-based hologram filter networks, in general, outperform the Swin Transformer based networks. This result is not unexpected. In [131], a large vision transformer underperforms models with ResNet backbone when a small dataset is utilized in training. Due to the

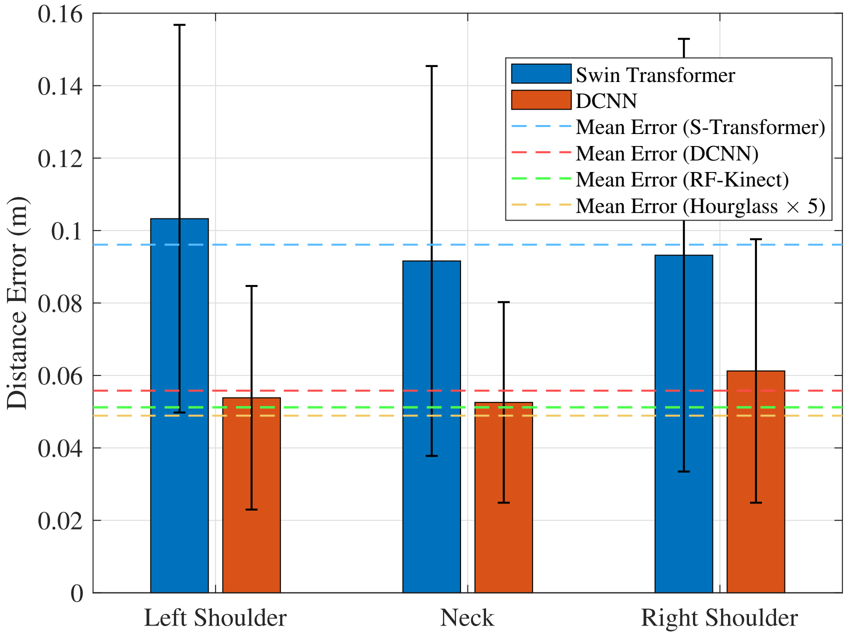


Figure 4.11: Location estimation for tags

fact that our dataset for indoor localization only has three hundred groups of input tensors, it is acceptable for us to achieve comparable location precision with a Swin Transformer based network. Furthermore, the interpretable filtered result is the main reason for us to investigate the Swin Transformer based network. In accordance with our observation in Fig.4.1 that peaks would always locate on a highlight band, Fig.4.8(c) recovers the real peak based on the band pattern in Fig.4.8(a), which meets our expectation in location estimation. Additionally, large vision transformer models overtakes ResNet based model as the dataset grows in the computer vision tasks. It shows us the potential of Swin Transformer based hologram filter network.

Fig. 4.11 depicts the mean distance errors for tags as well as the overall average errors of various network configurations. Apparently, the DCNN-based hologram filter network beats the network with a Swin Transformer backbone in terms of accuracy. The distance error obtained from the left shoulder with the Swin Transformer backbone is approximately $0.103m$, which doubles the error achieved with the DCNN backbone of $0.053m$. The lowest error is obtained from the neck in both networks, where the network with the DCNN backbone achieves $0.0525m$ and a error of $0.0916m$ is acquired by the Swin Transformer based network. In Fig. 4.11, the mean distance errors for various systems are also outlined with dashed lines. The blue and red lines show the overall distance errors for networks using the Swin Transformer

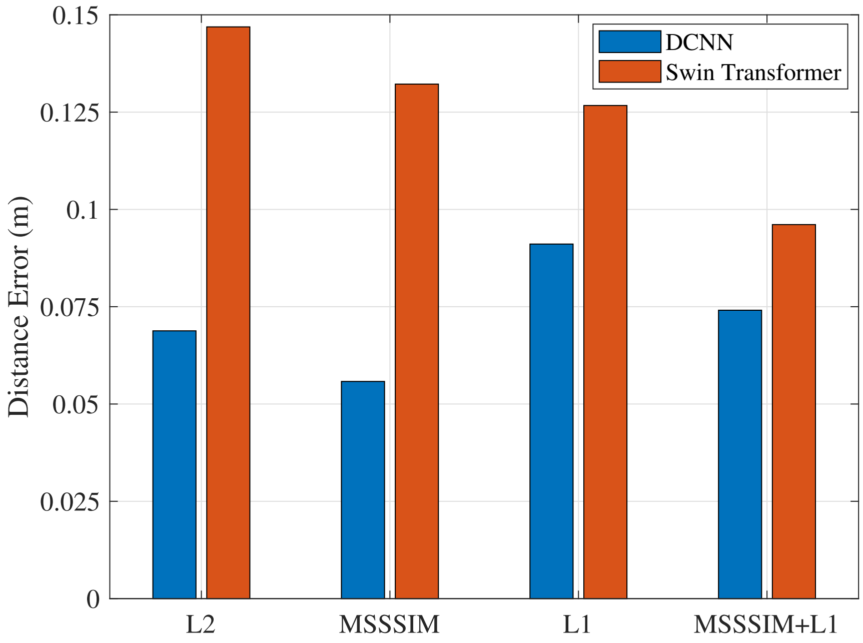


Figure 4.12: Location estimation effected by different loss function

and DCNN backbones, respectively. Because RF-Kinect [106] also performed an experiment in a $1.5m \times 1.5m$ scanning region, its average distance error of $0.0512m$ is displayed with a green line for comparison with our proposed approaches. As illustrated in Fig. 4.11, even though RF-Kinect exhibits a small improvement in localization accuracy when compared to the DCNN-based network, the extended version of the DCNN-based hologram filter network, denoted as Hourglass×5, achieves an error of $0.0489m$, outperforming all other methods. The network extension would be discussed in the following paragraphs.

Two representative deep neural networks are used as backbones in this chapter to sanitize the noisy hologram tensors. First, the framework employs a 3D variant of the U-Net with a Swin Transformer backbone. This type of network is proposed in [144] for semantic segmentation. However, the purpose of semantic segmentation is to label each pixel in the image with the right label, which does not match our task of tensor sanitizing. Labels will not present in our task. The pixels are filtered with the network to produce a smooth and continuous sanitized tensor. Tensor sanitizing, from this perspective, is similar to image restoration and denoising. The fake peaks related to phase wrapping and multipath could be treated as blur and noisy in the hologram tensor. Therefore, two loss function in image restoration, L1 loss and MSSSIM loss [138], are introduced to the Swin Transformer based network for enhancing the performance

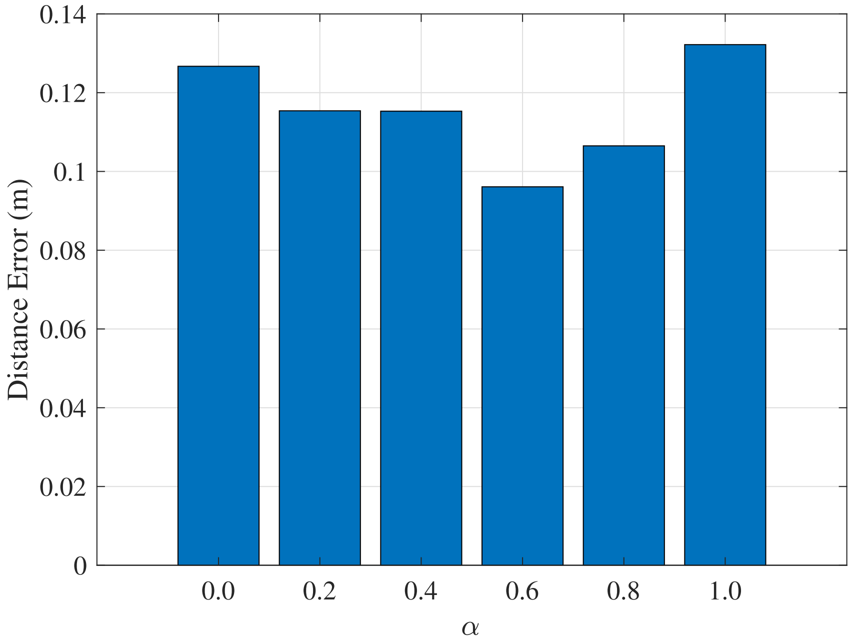


Figure 4.13: Location estimation effected by α using Swin Transformer backbone

of tensor sanitizing. According to Fig. 4.12, the mean distance error is $0.1469m$ when L2 loss is utilized in the training. The localization estimate continues to improve with the use of MSSSIM loss and L1 loss. The MSSSIM loss reduces the mean distance error to $0.1322m$, while the L1 loss contributes to a mean distance error of $0.1267m$. Moreover, we used a joint loss function that included both MSSSIM loss and L1 loss to improve the performance of the Swin Transformer based hologram filter network. When self-supervised learning is used, the distance error is optimized to $0.0961m$. Since the joint loss function displays outstanding performance in improving the localization accuracy for Swin Transformer based hologram filter network, we explore the effect of the ratio between L1 loss and MSSSIM loss on the distance error in Fig. 4.13. It is evident that the distance error stays high when L1 loss, or the MSSSIM loss, is deployed in the training individually, which are displayed when α is 0.0, or 1.0. With the increment of α, MSSSIM loss is introduced into the supervised training. The distance error drops to $0.1154m$ when α is 0.2. Even though a slight stagnation happens as α is 0.4, the lowest distance error achieves when α rise to 0.6. After then, as the L1 loss disappears, the localization accuracy continues to deteriorate.

However, the DCNN-based hologram filter network does not benefit from the the joint loss function built of L1 loss and MSSSIM loss. The hourglass network, a convolutional network

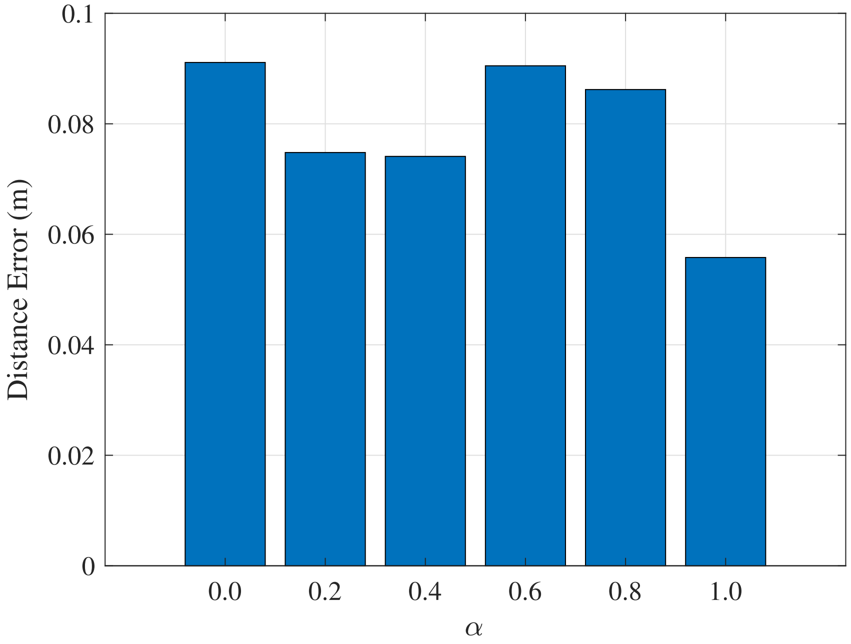


Figure 4.14: Location estimation effected by α using DCNN backbone

architecture for human pose estimation, serves as the backbone of the DCNN-based hologram filter network. The core idea of the hourglass network is to capture the spatial interactions associated with the key points using a repeated bottom-up, top-down covolutional structure. It converges to our tensor filter task, in which we attempt to extract the true peaks existing in a multiple-channel hologram tensor by using the spatial relationship between peaks in different channels. However, the loss functions for image restoration could not match the hourglass network perfectly. L1 loss produces the worst distance inaccuracy of $0.0911m$ in Fig. 4.12. Despite the fact that MSSSIM loss leads to the best localization accuracy with a mean distance error of $0.0558m$, the joint loss function could not replicate its effect on the Swin Transformer backbone. The distance error related to L1 loss is even higher than the error achieved by L2 loss, which are $0.0741m$ and $0.0688m$, respectively. In Fig. 4.14, we also investigate the effect of the ratio α between L1 loss and MSSSIM loss on the distance error. Although the distance error drops as MSSSIM loss involves in the training, the overall localization accuracy is not elevated by the joint loss function. When α is 0.6, the localization accuracy even gets worse.

To acquire the best localization accuracy using the DCNN-based hologram filter network, a joint loss function which consists of L2 loss and MSSSIM loss is studied. β is the ratio between L2 loss and MSSSIM loss, which follows the similar way in (4.10). The effect of β on

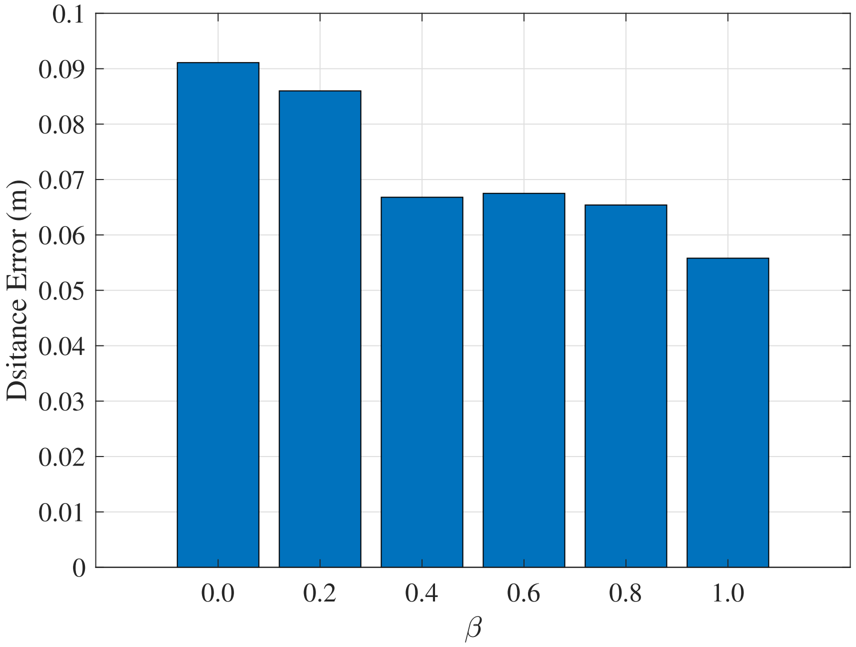


Figure 4.15: Location estimation obtained by the joint function built of L2 loss and MSSSIM loss

Structure	#param.	Distance Error (m)
(2,2,2,2)*	62.2M	0.0961
(2,2,4,2)	63.1M	0.0985
(2,2,6,2)	64.1M	0.1072
(2,2,8,2)	65.0M	0.1064

Table 4.3: Location estimation effected by the setup of the Swin Transformer based hologram filter network. (* denotes the default setting)

the distance error is illustrated in Fig. 4.15. With the growth of β, MSSSIM loss is utilized to advance the localization accuracy. As β reaches 0.4, the MSSSIM loss significantly improves the situation, but after that the growth rate slows. Eventually, the optimized distance error is obtained when the loss function is composed by MSSSIM loss individually.

The backbone architecture would have a considerable impact on the performance of the hologram filter networks. To evaluate the localization accuracy resulted by the structural changes, we conduct experiments with different number of trainable parameters using two hologram filter networks. In Table 4.3, the number of layers in a Swin Transformer based network is first modified. According to [110], only the layers in the stage-3 is modified, where $(2, 2, 2, 2)$ indicates a default layer configuration that two layers are included in each stage.

Dimension Size	#param.	Distance Error (m)
12	4.07M	0.1175
24	15.7M	0.1135
48*	62.2M	0.0961

Table 4.4: Location estimation effected by the feature size. (* denotes the default setting)

Structure	#param.	Distance Error (m)
Hourglass×2	33.4M	0.0565
Hourglass×3*	49.9M	0.0558
Hourglass×4	66.3M	0.0523
Hourglass×5	82.7M	0.0489

Table 4.5: Location estimation effected by the setup of the DCNN based hologram filter network. (* denotes the default setting)

As we can see, the number of parameters grows with the increasing number of layers. However, the distance error does not improves significantly. Despite an increase in the number of parameters from 62.2M to 65.0M, the distance inaccuracy remains about $0.1m$.

Another key parameter influencing the scale of the Swin Transformer backbone is dimension size, given as dim in Table 4.2. It determines the output dimension of the linear layer in a transformer block. Because the dimension of the current stage is determined by the dimension of the preceding stage, any change in the first stage would drastically alter the scale of the entire network. Table 4.4 employs three different dimension sizes, 12, 24, and 48, to investigate their impact on the number of parameters and localization accuracy. When the dimension size is 12, the network consists of just 4.07M trainable parameters; therefore, the capability of the network is constrained by the confined size. The distance error degrades to $0.1175m$. As we double the dimension size to 24, the number of trainable parameters inflates to 15.7M. Correspondingly, the extend of the network size contributes to the enhanced localization accuracy. The distance error drops to $0.1135m$. We further increase the dimension size to 48 to inspect the improvement in localization accuracy brought by the enlarged network. In this scenario, the hologram network is composed of 62.2M parameters and the distance error achieves $0.0961m$.

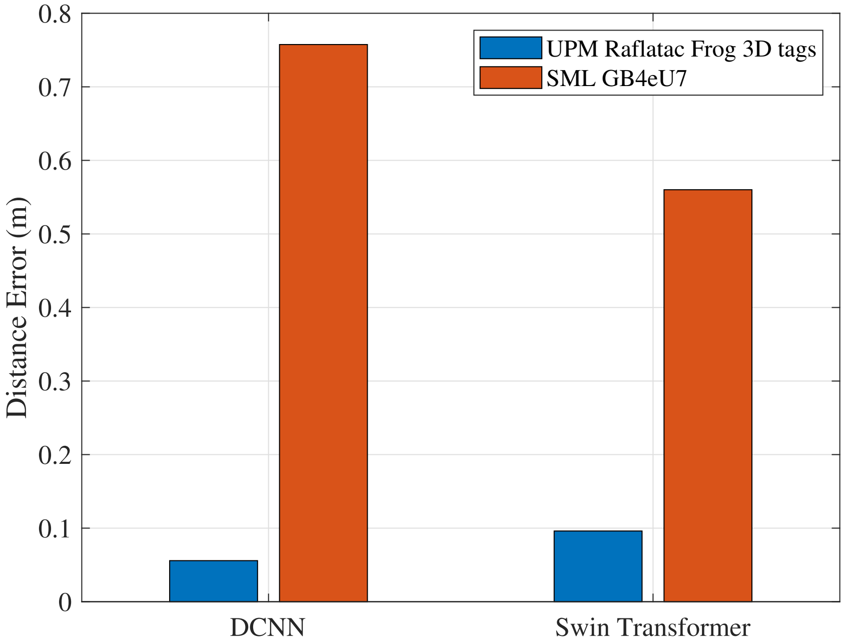


Figure 4.16: Location estimation obtained from different tags

We also study the influence of network size on the localization precision of the DCNN-based hologram filter network. Due to the architecture of the hourglass backbone, it is convenient to stack several hourglass units for network expansion. Four variations of the hourglass backbone are deployed in Table 4.5. The number of parameters grows in direct proportion to the number of hourglass units leveraged in the backbone. With the increment of the number of parameters, the distance error declines gradually. The lowest distance error of $0.0489m$ is obtained when five hourglass units, including 82.7M parameters, are used in the network.

To investigate the robustness of the hologram filter networks, SML GBe4U7 tags are deployed in the framework to collect the hologram tensors. The tags are attached to the human body in the same position as in the previous experiment. The newly collected tensors would not be used to train, or fine tune, the hologram filter networks. Using previously trained networks, the experimental results are obtained with the newly collected tensors directly. Fig. 4.16 delineates the performance degradation resulted from different tags. Obviously, a significantly performance drop occurs to both networks. However, the Swin Transformer based network is more robust to the change of tags. Even though DCNN-based network reaches the distance error of less than $6cm$, its accuracy suffers when facing tags that have never participated in training. The Swin Transformer based network leads the DCNN network by about $20cm$ in the new tag test. It potentially shows that the interpretable pattern extracted by the Swin Transformer based network is beneficial to tensor sanitation in different tags, whereas the DCNN based network could possibly clean the tensors through some ”shortcuts”, which hampers its transferability.

4.6 Conclusions

In this chapter, we provide MulTLoc, a framework that utilizes deep neural networks for RF hologram tensor filtering in order to locate multiple RFID tags. To our knowledge, this is the first paper to utilize hologram tensors to train deep neural networks for RFID tag-based three-dimensional indoor localization. Two representative deep learning models are implemented with the MulTLoc framework. First, we built a DCNN-based hologram filter network. The network successfully recovers the cleaned hologram tensors. The centimeter-level multiple tag localization is achieved successfully with the sanitized hologram tensor. In addition, a Swin Transformer based network is also used to sanitize the hologram tensors for expand the compatibility of the proposed framework. The network architecture is not related to the number of target tags. By adopting self-supervised training, the network is effectively trained with a small dataset including a limited number of training tensors. We evaluate the proposed framework using a task of multiple-joint location estimation. The results demonstrate the outstanding performance of the suggested framework.

Chapter 5

AdvLoc: Adversarial Deep Learning for Indoor Localization

5.1 Introduction

Location-based services have drawn significant attention driven by the increasing popularity of Internet of Things (IoT) devices and applications for Global Positioning System (GPS) denied indoor environments. Emerging indoor localization systems adopt various radio frequency (RF) signals, such as WiFi, RFID, and Bluetooth, etc. [15, 17, 27, 28, 109, 134]. Among these, the WiFi signal has been dominant in such systems that provide location estimation for the indoor environment in people's daily life, because of its omnipresence and lower cost.

Traditionally, indoor localization systems rely on signal processing techniques to estimate the distance between a transmitter and receiver, the Angle-of-Arrival (AoA), or the Time-of-Flight (TOF), for inferring the target location. For example, SpotFi [112] utilized a modified MUltiple SIgnal Classification (MUSIC) algorithm to achieve decimeter-level location accuracy by using AoA and ToF. Chronos [145] was able to compute the sub-nanosecond ToF and estimate the target location with decimeter-level accuracy as well. However, these techniques are limited by the quality of the signal. In the indoor environment, WiFi signals are scattered and reflected by walls and furniture, which result in the inevitable noisy WiFi measurements, especially the phase readings. To alleviate the negative effect contributed by the offsets, indoor localization systems usually employ powerful but time-consuming algorithms, such as the super-resolution algorithm used in SpotFi, which limits their performance for realtime applications.

Deep learning has been a hot topic since it has achieved great success in solving tasks, such as data compression, speech recognition, and image classification. Recently, indoor localization systems also benefit from the development of deep learning. Compared with traditional systems, deep learning makes such systems more efficient in location estimation, even though it would take more time for training the model. The first work applying deep learning to indoor localization is DeepFi [27], which leverages a stack of Restricted Boltzmann Machines (RBMs) to build an autoencoder for extracting location features from WiFi Channel State Information(CSI). PhaseFi [28] and BiLoc [29] further improve the the location accuracy by leveraging different CSI data. Due to the fingerprinting method, the localization problem is transferred to a matching problem. In the training stage, autoencoders have to be trained at each training location for extracting fingerprints. The training process could be time-consuming and the size of fingerprint data may restrict the deployment of the localization system in mobile devices which usually have limited storage. To overcome the drawbacks of the autoencoder based localization systems, CiFi [109] is the first work to utilize Deep Convolutional Neural Networks (DCNN) for indoor localization. With DCNN, location estimation is treated as a multi-class classification problem. Thus, the localization system only needs to train one DCNN model in the training process, and the fingerprints collected in the training stage are not essential for location estimation once the DCNN is trained successfully. Like CiFi, Received Signal strength (RSS) and CSI amplitude have also been utilized to train the DCNN model [61, 81, 146, 147]. ResLoc [108, 123] proposed a sharing learning approach based on deep residual learning, which uses the bimodal CSI tensor data.

Even though deep neural networks (DNN) have achieved excellent performance on classification problems, some counter-intuitive properties of DNNs have also been exposed along with its popularity. Szegedy et al. [148] found that several machine learning models, including state-of-the-art neural networks, are vulnerable to adversarial examples. Goodfellow [149] verified the discovery by misleading the GoogLeNet [150] with adversarial examples. Deep learning based indoor localization systems also face the threat of adversarial attacks. To evaluate and counteract the threat of adversarial attacks to DNN-based indoor localization systems, we propose **AdvLoc**, an **Adv**ersarial deep learning for indoor **Loc**alization system. Like traditional DCNN based systems, AdvLoc operates in two stages, an offline training stage and an online location estimation stage. We apply adversarial attacks in the online stage, where the perturbations generated by adversarial attacks are be introduced to the existing clean inputs of the DCNN. In the offline stage, the DCNN based localization model will be trained adversarially to enhance its robustness against the adversarial examples. Unlike the image classification models, the DCNN model in the indoor localization system process the online inputs that do not belong to any existing class in the training dataset (i.e., the mobile device may be placed at an arbitrary location, rather than a known training locations). Using the AdvLoc system, we evaluate the effects of six types of mainstream adversarial attacks on DCNN based indoor localization with respect to accuracy and location error. To defend against such attacks, adversarial training is implemented in the offline training of the models. The experimental results validate that adversarial training utilized in the proposed AdvLoc system is an effective means to counteract the location errors cause by the first-order adversarial attacks.

The main contributions made in this chapter can be summarized in the following.

- We expose the threat of the adversarial attacks to deep learning based indoor localization systems by visualizing the adversarial examples and evaluating the impact of the various magnitude of the perturbation on the adversarial examples to the location estimation. The effect of six types of representative adversarial attacks, including gradient-based, optimization based, and spatial transformation based attacks, on the indoor localization system, is investigated in both white-box and black-box attack scenarios.

- To the best of our knowledge, this is the first work to employ adversarial training to enhance the robustness of WiFi CSI-based indoor localization systems. We introduce adversarial training into the traditional DCNN based indoor localization. In the white-box attack scenario, the modified loss function successfully alleviate the negative effect resulted from the first-order adversarial attacks, especially Fast Gradient Sign Attack (FGSM) [149].

- The proposed AdvLoc system is implemented with commodity 5 GHz WiFi. We verified its performance in two representative indoor environments with extensive experiments. The experimental results exhibit the threat of adversarial attacks and show that adversarial training effectively improves the robustness of the localization system when the input examples are manipulated by first-order adversarial attacks.

The remainder of this chapter is organized as follows. Section 5.2 reviews related work. We present the AdvLoc design in Section 5.3 and our experimental study in Section 5.4. Finally, Section 5.5 concludes this chapter.

5.2 Related Work

With the advances in computing power, the availability of data, and the development of open-source platforms, deep learning has been recognized as a powerful tool for many real world problems that cannot be solved by conventional machine learning techniques. However, as Szegedy et al. first unveiled in [148], using image classification as an example, the resilience of deep learning has been exposed to the threat of adversarial attacks. Nowadays, most AI-based services, such as Apple Face ID and Amazon Alexa, are highly dependent on the progress of deep learning in image classification and Natural Language Processing (NLP). The vulnerability of deep learning networks place user privacy and public safety at risk.

Following the discovery in [148], Finlayson et al. [151] investigated the vulnerabilities of the medical AI systems under adversarial attacks and pointed out that the adversarial attacks may already be in place and contribute to medical fraud. The diagnostic performance could be affected easily by adding a small perturbation generated by the common adversarial attacks, while the manipulated diagnostic probability could deceive the automated fraud detector evaluating the medical claims. Furthermore, Finlayson et al. also indicated that the adversarial attacks are effective for extremely accurate medical classifiers even if the prospective attackers do not have access to the deep learning model. In [152], both white-box and black-box Projected Gradient Descent (PGD) attacks were used to generate adversarial examples. The result showed that state-of-the-art medical models were misled in both scenarios. Furthermore, researchers have applied adversarial attacks in other real-world scenarios. For example, Thys et al. [153] proposed an approach to generate adversarial patches to hide a person from a DCNN based human detector. Sharif et al. [154] presented an approach for generating eyeglass frames to fool state-of-the-art Face Recognition Systems (FRSs). The experimental results showed that their techniques were effective for black-box FRSs, as well as state-of-the-art face detection systems (FDSs).

Not only traditional DCNNs but also the Spatio-Temporal Graph Convolutional Network (ST-GCN) is facing the threat of adversarial attacks. Unlike the medical AI systems relying on DCNN for image classification, action recognition applications utilizing ST-GCN for processing the skeleton data obtained from RGB-D sensors [155, 156]. Liu et al. [157] proposed Constrained Iterative Attacks for Skeleton Actions (CIASA), which was based on FGSM and was able to disturb the joint locations in an action sequence. Even though the features of graph nodes and graph structure were discrete with certain predefined structures, the basic FGSM attack was able to fool the ST-GCN in the form of non-targeted attacks.

Even though the textual data is different from image data composed of continuous pixel values, adversarial examples affect DNN for text-based tasks as well. Three types of perturbation strategies, namely insertion, modification, and removal, were introduced in TextFool [158] based on the concept of FGSM. In [159], authors showed that the recurrent neural network (RNN) is not immune to the adversarial attacks. The attack methods used in crafting adversarial image examples could be adapted to generate sequential adversarial text by leveraging computational graph unfolding. In a recent work [160], we investigated the problem of adversarial attacks on solar power generation forecasting, which is a regression problem, and showed that both DNN and a LASSO-based statistical model were vulnerable.

Recnetly, there has been considerable interest of applying deep learning to wireless communications and networking problems [161, 162]. Because adversarial attack has been a common threat to deep learning systems, researchers have also investigated the impact of adversarial attacks in wireless systems. For example, modulation recognition, is a key technology of Cognitive Radio (CR), for which deep learning techniques have been developed. In [163], Sadeghi et al. demonstrated how adversarial examples degrade the model performance of radio signal (modulation) classification. Compared with traditional attacks such as jamming, the adversarial attack required much less power since only small perturbations were generated. Lin et al. [164] evaluated four representative adversarial attacks on modulation recognition. The results showed that, regardless of white-box or black-box, adversarial attacks could reduce the accuracy of the target model, while the performance of iterative attacks was superior to that of single step attacks. A thorough study of adversarial attacks on IoT device identification (or, device fingerprinting) was reported in [165], which was to identify specific wireless transmitters based on received signals. Although following the same specifications and using the same protocols, the devices can still be distinguished by the small defects incurred during the manufacturing process or the aging process.

5.3 The AdvLoc System

Due to the popularity of mobile devices, location information has been an essential part of IoT. Recently, an increasing number of researchers have focused on WiFi-based indoor localization because of the ubiquitous availability and low cost of WiFi devices. Many indoor localization systems [27–29] rely on the fingerprinting method, which means the fingerprints of known locations need to be measured and stored in a database for online localization. To reduce the storage requirement, some systems [61, 109] treat indoor WiFi fingerprinting as a classification problem, where DCNN becomes the best choice owing to its great success in image classification. As Szegedy et al. [148] revealed the vulnerability of DCNN models to adversarial examples, consequently, the DCNN based localization systems would also be susceptible to adversarial attacks. To combat such threats, we propose the AdvLoc system in this chapter, which utilizes adversarial training in the offline stage to enhance the robustness of the network, making it immune to adversarial examples.

5.3.1 Architecture of the AdvLoc System

Fig. 5.1 depicts the architecture of the proposed AdvLoc system. Like traditional DCNN based indoor localization systems, AdvLoc comprises of an offline stage and an online stage. In the

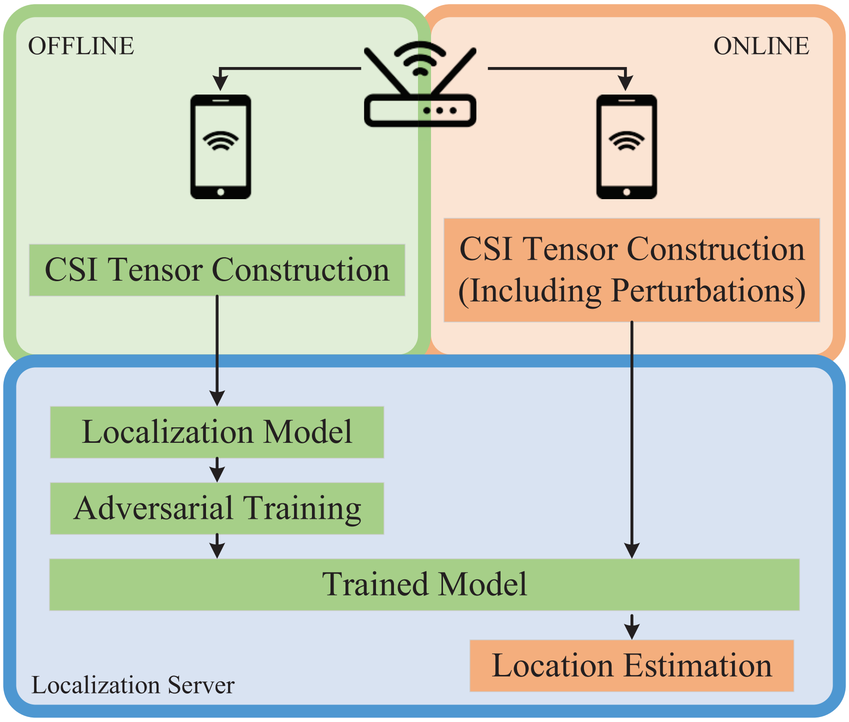


Figure 5.1: Indoor localization architecture.

offline stage, CSI tensors are constructed using the CSI data collected at the receiver for the mobile device placed at various known training locations. The core of AdvLoc is a deep residual learning model, ResNet [9], that learns location features from WiFi CSI data. The ResNet will be trained adversarially using CSI tensors to generate the model for online localization. The newly received CSI data is collected from mobile devices placed at an unknown location in the online stage. The adversarial perturbations are generated and injected in CSI tensor in the online stage, even though the new CSI tensors are constructed in the same way as in the offline stage. As a result, the wireless channel has no effect on the perturbations inserted into the CSI tensors. Moreover, because adversarial attacks happen at the process of CSI tensor generation, it is more feasible in both white-box and black-box scenarios.

Specifically, in the offline stage, the training dataset and verification dataset are collected from identical positions. The collected observations, such as phase readings, are labeled by the coordinates of corresponding positions. The location with the highest similarity in the output of the DCNN model is selected as the output of the system. Therefore, we could assess how well our model fits the training data using the verification dataset by examining the verification accuracy in this stage. In the online stage, the testing dataset is collected from the positions not used in the offline stage. Obviously, the classification accuracy would not be persuasive to demonstrate the localization performance of the system. In fact, the output of ResNet is used as the similarity to calculate the estimated location. The estimated location, $\widehat{T}$, is computed by

$$\widehat{T} = \sum_{i=1}^{N} t_i \times p_i \tag{5.1}$$

where p_i is the output of the ResNet that depicts the similarity between the testing location and the training location i, and t_i is the known training location i.

5.3.2 CSI Tensor Construction

The CSI tensor used in AdvLoc consists of three slices. Two of the slices are generated with the estimated angle-of-arrival (AoA) values using the phase difference data from the three receiving antennas, while the third slice contains the measured CSI amplitude values. Considering that the Intel WiFi Link 5300 network interface card (NIC) only supports 3 antennas and 30 subcarriers for each antenna, the size of the CSI tensor is set to $30 \times 30 \times 3$. Fig. 5.2 depicts CSI tensors used in our AdvLoc system when different levels of perturbations are introduced (as indicated by the parameter ϵ). As we can see, when $\epsilon = 0$, no perturbation is added and the tensor is a clean input to the ResNet model. Whereas, the rest of the tensors are adversarial examples generated using the FGSM method, where ϵ is a hyper-parameter that controls the magnitude of the perturbation. When ϵ is less than 0.4, the perturbation added in the tensors is negligible (i.e., visually invisible). However, the tensor will be distorted obviously, once ϵ is larger than 0.5. We shall study the relationship between ϵ and the location estimation error in the following sections.

5.3.3 Architecture of the ResNet Models

To investigate the effect of adversarial attacks on DCNN-based indoor localization systems, two popular ResNet models are adopted in the AdvLoc system, including ResNet-18 and ResNet-50 [9]. The ResNet-18 model will be leveraged as the localization model in our study of both

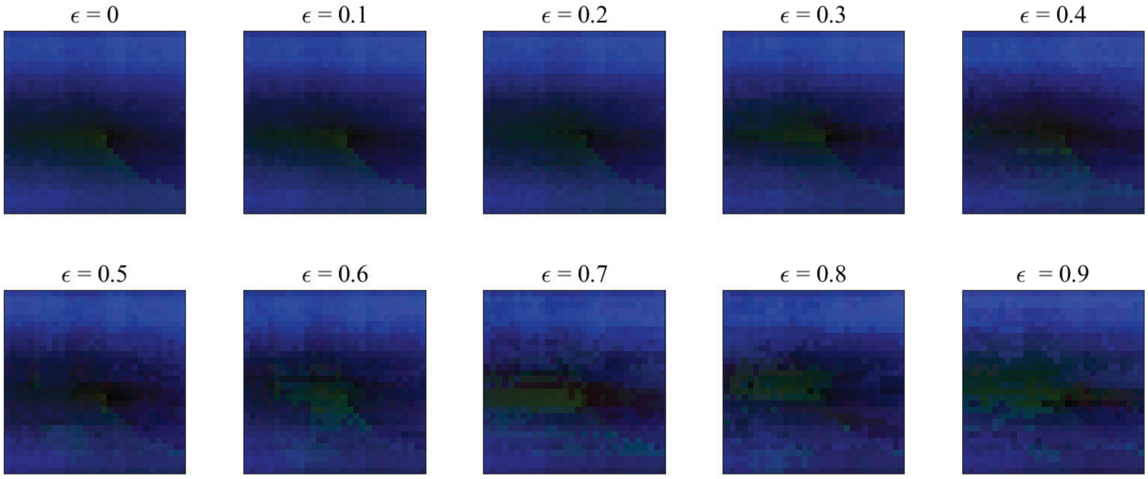


Figure 5.2: Examples of CSI Tensors when different levels of pertubations are introduced, as indicated by the hyper-parameter ϵ.

	Input Block	Conv_Block1	Conv_Block2	Conv_Block3	Conv_Block4	Output Block
ResNet-18	$\begin{bmatrix} 7\times 7, 64 \\ max\ pool \end{bmatrix}$	$\begin{bmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} average\ pool \\ fully\ connected\ layer \\ softmax \end{bmatrix}$
ResNet-50	$\begin{bmatrix} 7\times 7, 64 \\ max\ pool \end{bmatrix}$	$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512 \\ 1\times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} average\ pooling \\ fully\ connected\ layer \\ softmax \end{bmatrix}$

Figure 5.3: Architecture of the two ResNet models used in AdvLoc: ResNet-18 and ResNet-50 [9].

white-box attacks and black attacks. In the study of black-box attacks, the ResNet-50 model will be trained as a substitute model for mimicking the localization model, i.e., the ResNet-18 model.

Fig. 5.3 exhibits the detailed structure of the localization models. The building units shown in the brackets depict the component of each block. For example, the input block of the ResNet-18 model includes 7x7 filters for generating 64 feature maps, then the max pooling is leveraged to shrink the size of the feature maps. As for the Conv_Block4 of the ResNet-18 model, it is composed of two building units. In each unit, it contains two 3x3 convolution layers. The shortcut connection exists in each building unit of Conv_Block. Since the localization problem is treated as a classification problem in the fingerprinting based localization system, the cross-entropy loss is utilized in the training process.

5.3.4 Adversarial Attacks

Szegedy et al. in [148] showed that adversarial examples hardly distinguishable from the originals can fool DCNN-based image classifiers such as AlexNet [166]. Since then, security has become an important problem in AI/ML research, especially for privacy-sensitive applications such as localization. To better evaluate the resilience of DCNN-based localization systems against adversarial attacks and the effectiveness of defense strategies, we implement the following six types of adversarial attacks in this study.

Fast Gradient Sign Method (FGSM)

FGSM was proposed by Goodfellow et al. in 2015 [149]. The method obtains a perturbation, denoted by η, by calculating the gradient of the loss function $L(\cdot)$ with a given input, as

$$\boldsymbol{\eta} = \epsilon \cdot \operatorname{sign}(\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}, y)), \tag{5.2}$$

where θ represents the parameters of a well-trained model; $\boldsymbol{x}$ and y are the input and its corresponding label, respectively; ϵ is a hyper-parameter, which controls the magnitude of the perturbation. Since $L(\cdot)$ is the loss function of the model, the perturbation η can be calculated by using the first derivative of $L(\theta, \boldsymbol{x}, y)$ through the backpropagation algorithm.

In 2017, Goodfellow et al. [167] modified FGSM by cancelling the sign$(\cdot)$ function in (5.2). The new method, Fast Gradient Method (FGM), is a generalization of FGSM, where the perturbation is give by

$$\boldsymbol{\eta} = \epsilon \cdot \frac{\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}, y)\|_2}. \tag{5.3}$$

With (5.3), the perturbation can be easily created. However, it is not safe to say that the perturbation will contribute to misclassification, even though the loss value for the target label to be misclassified is increased by introducing the perturbation.

Projected Gradient Descent (PGD)

Based on the one-step FGM, an iterative version of FGM termed PGD, was proposed in 2017 [168]. Madry et al. created the PGD adversary to enhance the robustness of the classifier against the first-order attacks.

With the iterative method, the adversarial examples $\{\boldsymbol{x}_0^{adv}, \boldsymbol{x}_1^{adv}, ..., \boldsymbol{x}_{N+1}^{adv}\}$ are generated as follows.

$$\begin{aligned} &\boldsymbol{x}_0^{adv} = \boldsymbol{x}, \\ &\boldsymbol{x}_{N+1}^{adv} = \mathrm{Clip}_{x,\epsilon}\left\{\boldsymbol{x}_N^{adv} + \alpha \cdot \frac{\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x})}{\|\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}, y)\|_2}\right\}, \end{aligned} \tag{5.4}$$

where α is a hyper-parameter for each iteration, which is usually set as ϵ/N for a given ϵ. With this approach, the perturbation is always small and around the original input $\boldsymbol{x}$ in the L^p ball. Also, $\mathrm{Clip}_{x,\epsilon}$ is used to project the perturbation back into the L^p ball if necessary. PGD has been verified to be a stronger adversarial attack method than the one-step FGM/FGSM at the cost of transferability.

Momentum Iterative Method (MIM)

Since PGD generates adversarial examples with a greedy approach along the direction of the gradient in each iteration, the local maxima could be reached easily, resulting in poor transferability. To solve this problem, the momentum-based method is integrated into FGSM. Instead of using the gradient in one iteration to update the perturbation, the Momentum Iterative Method (MIM) leverages the gradient of the previous iterations to guide the update of the perturbation [169]. The memory of previous gradients can help to avoid the local maxima, which occur in PGD. Thus, it breaks the dilemma of choosing between the "underfitted" FGSM and the "overfitted" PGD.

To generate adversarial examples with MIM, we have

$$\begin{cases} \boldsymbol{g}_0 = 0, \\ \boldsymbol{x}_0^{adv} = \boldsymbol{x}, \end{cases} \\ \begin{cases} \boldsymbol{g}_{N+1} = \mu \cdot \boldsymbol{g}_N + \frac{\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}_N^{adv}, y)}{\left\| \nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}_N^{adv}, y) \right\|_1}, \\ \boldsymbol{x}_{N+1}^{adv} = \boldsymbol{x}_N^{adv} + \alpha \cdot \text{sign}(\boldsymbol{g}_{N+1}). \end{cases} \tag{5.5}$$

Note that $\boldsymbol{g}_N$ includes the gradients from previous $(N-1)$ iterations with a decay factor of μ. Here α can also be set to ϵ/N when ϵ is given. Thus, MIM retains the transferability of adversarial examples under increased iterations.

DeepFool Attack

In FGSM/FGM, the choice of the hyper-parameter ϵ significantly affects the performance of adversarial attacks, since ϵ decides the magnitude of perturbation. In DeepFool [170], perturbations are computed by solving optimization problems. For a binary affine classifier, $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$, the optimal perturbation is given by

$$\begin{aligned} \boldsymbol{\eta}^*(\boldsymbol{x}) &:= \text{argmin}\, \|\boldsymbol{\eta}\|_2 \\ \text{s.t. } \text{sign}(f(\boldsymbol{x_0} + \boldsymbol{\eta})) &\neq \text{sign}(f(\boldsymbol{x}_0)), \end{aligned} \tag{5.6}$$

which has the following closed-form solution

$$\boldsymbol{\eta}^*(\boldsymbol{x}) = -\frac{f(\boldsymbol{x_0})}{\|\boldsymbol{w}\|_2^2} \boldsymbol{w}. \tag{5.7}$$

The iterative method is adopted in DeepFool for general binary classifiers. In each iteration, Deepfool assumes f is linear in the neighborhood of the current $\boldsymbol{x}$. Hence the optimal perturbation is calculated as

$$\begin{aligned} \boldsymbol{\eta}^*(\boldsymbol{x}) &= \underset{\boldsymbol{\eta}_N}{\text{argmin}}\, \|\boldsymbol{\eta}_N\|_2 \\ \text{s.t. } f(\boldsymbol{x}_N) &+ \nabla f(\boldsymbol{x}_N)^T \boldsymbol{\eta}_N = 0. \end{aligned} \tag{5.8}$$

Considering that multi-class classification can be split into multiple binary classification, Deepfool could also find the optimized perturbation effectively for a non-linear multi-class neural network. Furthermore, it has been demonstrated that the adversarial examples generated by Deepfool have 5 times smaller perturbations comparing with those from FGSM on MNIST and CIFAR10 models.

Carlini Wagner Attack (CW)

Defensive distillation [171] is a popular defensive method, which robustifies neural networks to counteract adversarial examples. However, Carlini and Wagner proposed a type of attacks to make defensive distillation ineffective [172]. Among the various distance metrics used for evaluating similarities, the Carlini Wagner attacks (CW) are designed with the L_2, L_∞, and L_0 distance metrics. For the L_2 attack, adversarial examples are generated with $\boldsymbol{w}$, obtained by solving

$$\min \left\{ \left\| \frac{1}{2}(\tanh(\boldsymbol{w}) + 1) - x \right\|_2^2 + c \cdot f\left(\frac{1}{2}(\tanh(\boldsymbol{w}) + 1) \right) \right\}, \tag{5.9}$$

where the loss function $f(\cdot)$ is defined as

$$f(\boldsymbol{x}^{adv}) = \max\{\max\{\zeta(\boldsymbol{x}')_i : i \neq t\} - \zeta(\boldsymbol{x}^{adv})_t, -\psi\}, \tag{5.10}$$

where $\zeta(\cdot)_i$ is a logistic for class i, ψ controls the confidence with which the misclassification occurs, and c is a hyper-parameter that tradeoffs between the magnitude of perturbation and success rate of attack. For the L_0 attack, considering that the L_0 metric is non-differentiable, the pixels in $\boldsymbol{x}$ that affect the classifier significantly are selected and attacked with the Carlini and Wagner L_2 (CWL2) attack in an iterative manner.

To create adversarial examples with the L_∞ metric, the L_2 term in (5.9) is replaced by a penalty for any terms that exceed $\boldsymbol{\tau}$, i.e.,

$$\min \left\{ c \cdot f(\boldsymbol{x} + \boldsymbol{\eta}) + \sum_i \left[(\boldsymbol{\eta}_i - \boldsymbol{\tau})^+ \right] \right\}. \tag{5.11}$$

where $\boldsymbol{\tau}$ is decreased iteratively with an initial value of $\mathbf{1}$. Even though the CW attack has been demonstrated to have defeated the defensive distillation method, the time cost in generating adversarial examples using this method is much larger than that of all the previous attack methods.

Spatial Transformation Method (STM)

Unlike DeepFool and CW that construct adversarial examples by solving an optimization problem, the Spatial Transformation Method (STM) constructs adversarial examples with a natural transformation of the original inputs [173]. The transformation parameters, i.e., $(\delta_u, \delta_v, \theta)$, could be optimized by the grid search or the projected gradient descent method. The position of a pixel (u, v) is updated as follows.

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \delta_u \\ \delta_v \end{bmatrix}. \tag{5.12}$$

According to [173], STM can successfully defeat the CNN that was trained against an L_∞-bounded adversary.

5.3.5 White-box and Black-box Attacks

All the above attack methods are white-box attacks, which means that the adversary is capable of acquiring the knowledge of the target model, or even the training dataset. This possibility is usually slim in practice, especially for accessing the model and dataset related to personal privacy or homeland security. To make adversarial attacks more feasible, the more challenging black-box attacks have been investigated, where the attacker has no or limited knowledge of the model. We will also leverage black-box attack methods to evaluate the threat of adversarial attacks to the AdvLoc system.

A comparison of white-box and black-box attacks is shown in Fig. 5.4, where a substitute model is utilized to mimic the black-box model with infinite queries. Since information of the substitute model is open to the attacker, all of the attack methods designed for the white-box scenarios can be leveraged to fabricate adversarial examples in the black-box scenario. Due to the transferability of the adversarial examples, the black-box model would also be misled by the adversarial examples. However, this strategy is easy to be detected. Moreover, Papernot et al. [174] noticed that it will be intractable for attackers to build a substitute model with a limited number of queries. Thus, a Jacobian-based Dataset Augmentation technique (JAD) will be used in our AdvLoc system, which ensures that the substitute model is able to approximate the decision boundary of the black-box attack with a limited number of queries. Fig. 5.5 depicts the procedure of JAD. First, a small dataset D_0 is collected and labeled by the black-box model O. The substitute model will be trained with the dataset $(D_0, \widetilde{O}(D_0))$. Next, D_0 is augmented to generate a larger dateset D_1 given by

$$D_1 = \left\{ \boldsymbol{x} + \beta \cdot \text{sign}(J_F[\widetilde{O}(\boldsymbol{x})]) : \boldsymbol{x} \in D_0 \right\} \cup D_0, \tag{5.13}$$

where β is a parameter of augmentation, and J_F is the Jacobian matrix of the substitute model F. Thus, a growing augmented dataset will be generated iteratively and be leveraged to force the substitute model to approximate the black-box model. In this chapter, we would utilize JAD for all the previous attack methods to investigate the black-box attacks and defense for the indoor localization systems.

5.3.6 Position of Adversarial Attack

Because of the nature of the wireless communication systems, the adversarial attack is always launched in three targets, i.e., the receiving-side, the transmitting-side, and the channel-side. For the indoor localization systems, such as the WiFi based localization system, the APs, that are equipped in the indoor environment, play a role of transmitter. Since APs is an essential part of existing communication systems that are secured by cybersecurity technologies, it is challenging to inject perturbation through the transmitter (AP) side. On the other hand, the adversarial attacks from the channel-side is feasible because of the openness of the wireless channels. However, the channel effect has to be considered in the design of adversarial perturbations. For the advLov, we assume that the adversarial perturbations are injected when CSI

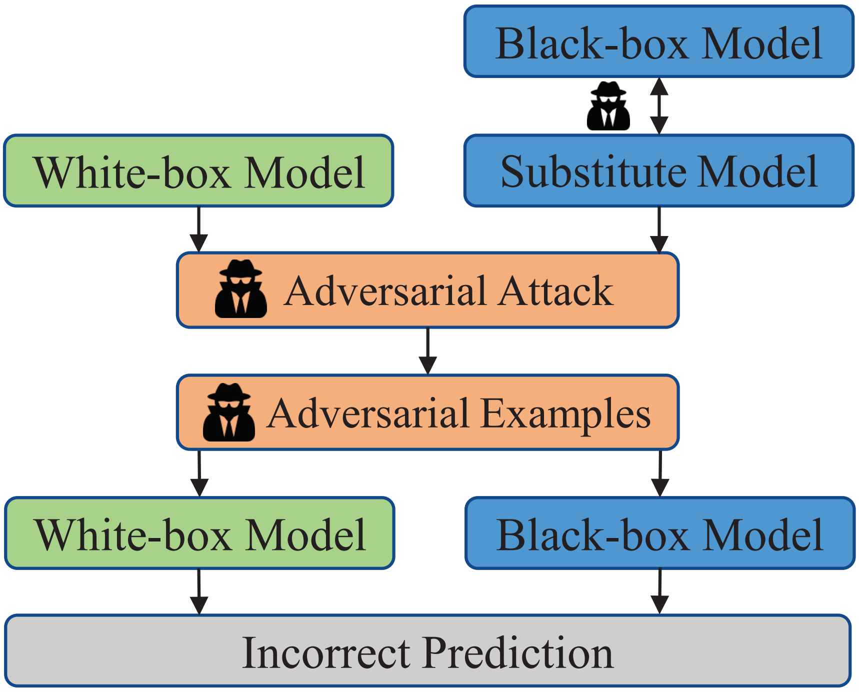


Figure 5.4: A comparison of white-box and black-box attack approaches.

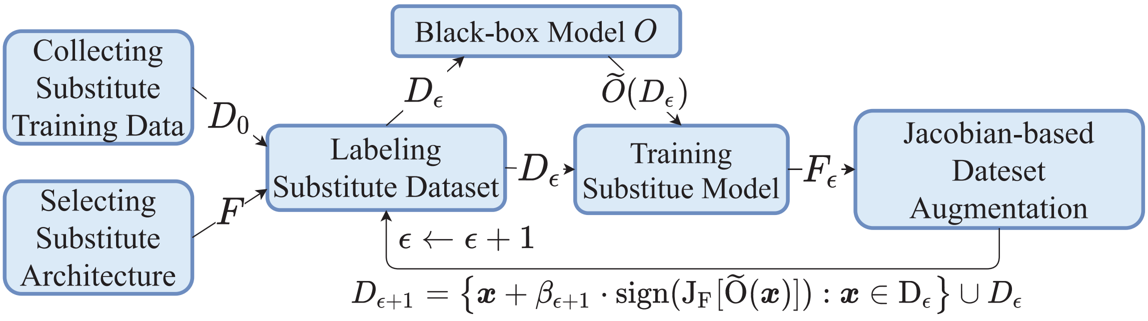

Collecting Substitute Training Data
Selecting Substitute Architecture
D_0
F
Black-box Model O
D_ϵ
$\tilde{O}(D_\epsilon)$
Labeling Substitute Dataset
D_ϵ
Training Substitue Model
F_ϵ
Jacobian-based Dateset Augmentation
$\epsilon \leftarrow \epsilon + 1$
$D_{\epsilon+1} = \left\{ \boldsymbol{x} + \beta_{\epsilon+1} \cdot \mathrm{sign}(\mathrm{J_F}[\tilde{\mathrm{O}}(\boldsymbol{x})]) : \boldsymbol{x} \in \mathrm{D}_\epsilon \right\} \cup D_\epsilon$

Figure 5.5: Training the substitute model.

tensor is generated, which is usually happens at the user side. Comparing with attacking transmitters (APs), the receive-side (user side) attack is more feasible because it is more possible for personal users to lack of sense in cybersecurity. Attackers could obtain the authority of victims with common methods, like phishing and malware, to inject the adversarial perturbations. Moreover, the channel effect is also eliminated when the perturbations are added in the user-side.

5.3.7 Adversarial Training

To make the AdvLoc system resilient to adversarial attacks, its localization model implements adversarial training, which enhances the robustness of the neural network by training it with a mixture of adversarial and clean examples. The basic idea of adversarial training is to augment the original loss function with an adversarial term, so that it will be resistant to adversarial examples. Goodfellow et al. [149] demonstrated that the adversarial loss function below

$$\widetilde{L}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \gamma \cdot L(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1 - \gamma) \cdot L(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\eta}, y) \tag{5.14}$$

was effective to make the neural network immune to FGSM attacks, where $\boldsymbol{\eta} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}, y))$. In (5.14), γ is a hyper-parameter to adjust the relative importance of the loss terms of the original and adversarial examples, which is set to 0.5 in our implementation of AdvLoc.

In the next section, we will leverage adversarial training to study the effect of defense for indoor localization systems against adversarial attacks. The resulting localization model that is adversarially trained will be called by the corresponding attack method used in adversarial training. For example, if the localization model is trained with loss function (5.14) and the disturbance $\boldsymbol{\eta}$ in (5.14) is generated using FGSM (or MIM and PGD), the resulting adversarially trained model will be called FGSM-AT (or MIM-AT and PGD-AT, respectively).

5.4 Experimental Study

5.4.1 Experiment Configuration

To evaluate the performance of AdvLoc under adversarial attacks in the online stage, we deploy the six types of adversarial attacks in both white-box and black-box scenarios. The AdvLoc system is implemented with Intel 5300 NIC in the 5.58 GHz band. Two laptops are configured as an access point and a mobile device, respectively. The distance between adjacent antennas is adjusted to 2.68cm, which is a half of the wavelength. To inject adversarial attacks in the online stage, CleverHans [175] is leveraged to generate adversarial perturbations for each new

CSI tensor. Furthermore, both the localization model trained in the offline stage and the adversarial example generation model used in the online stage are implemented with the TensorFlow framework on a NVIDIA RTX 2080 GPU.

For the sake of diversity, we examine the AdvLoc system in two representative indoor environments, i.e., a straight corridor and a computer laboratory.

- *Straight Corridor:* First, the AdvLoc system is deployed in a straight corridor in Broun Hall in the Auburn University campus. This indoor testbed covers an area of $8 \times 24\, m^2$, which includes the rooms on both sides of the corridor. As a typical indoor structure, the straight corridor is simple. Since there is no obstacles that result in complex scattering and reflection of WiFi signals, the Line-Of-Sight (LOS) path is the dominant component in this environment. As is shown in Fig. 5.6, the red squares represent the training locations in the offline stage, while the green dots denote the testing location in the online stage. The single access point is placed at the right end of the corridor in Fig. 5.6. The distance between consecutive training locations is $1.8\, m$.
- *Computer Laboratory:* Next, we assess the AdvLoc performance in a computer laboratory, which is also located in Broun Hall. Compared with the corridor, the computer laboratory is a cluttered environment. Most of the LOS paths of WiFi signals are blocked by tables, chairs, and computer chassis. In this case, the access point is placed close to the north center of the laboratory so that it could cover the entire area. Fig. 5.7 depicts the selection of training positions (marked as red squares) and testing locations (marked as green dots). The distance between adjacent training locations is also $1.8\, m$.

To evaluate the system performance, we investigate the verification accuracy in the offline stage (see Section 5.3.1). Because the training dataset and testing dataset are collected from identical locations, verification accuracy is defined as

$$\pi = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}, \tag{5.15}$$

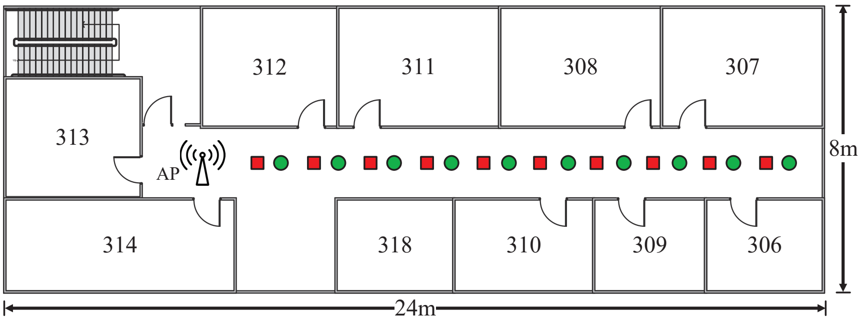


Figure 5.6: The layout of the corridor scenario.

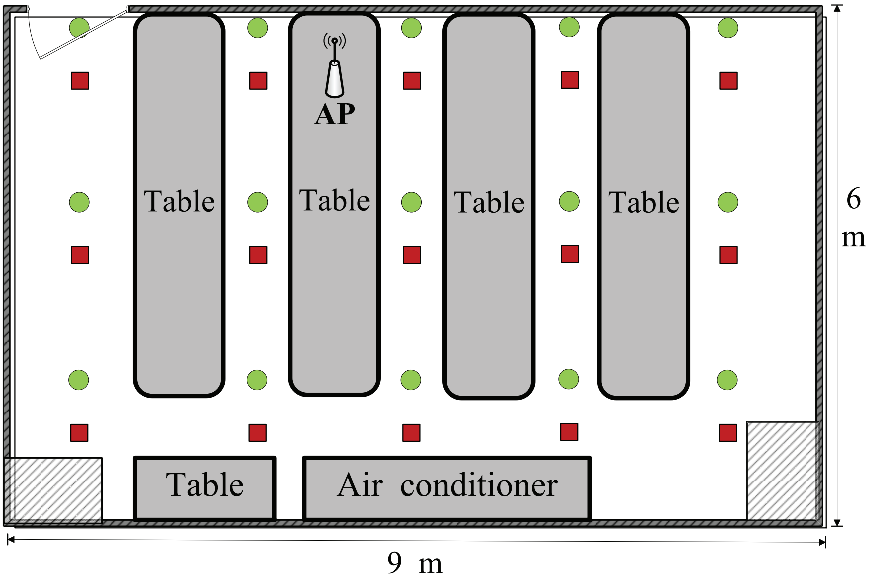


Figure 5.7: The layout of the lab scenario.

which indicates the capability of the DCNN model in solving the multi-class classification problem. In addition, we also evaluate the performance of the localization system by calculating the location estimation error $\mathcal{E}$, given by

$$\mathcal{E} = \|\widehat{T} - T\|_2, \tag{5.16}$$

where $\widehat{T}$ is the estimated location given in (5.1) and T is the ground truth.

5.4.2 Verification Accuracy Under White-box Attacks

We first confirm the verification accuracy of AdvLoc under white-box attacks in both indoor environments. For indoor localization systems, the training dataset and verification dataset are collected from identical positions. The verification accuracy gives us an unbiased assessment of how well our model fits the training data. Fig. 5.8 depicts the verification accuracy of the original localization model when not being attacked (called "Original Model"), and the verification accuracy of the original model when attacked by adversarial examples generated using FGSM, MIM, and PGD (called "Original Model (FGSM)," "Original Model (MIM)," and "Original Model (PGD)," respectively) in the lab setting. It shows that all the three attack methods successfully degrade the verification accuracy as ϵ is increased from 0.1 to 1. It is intuitive that a larger magnitude of perturbation causes a larger decrease in verification accuracy. Fig. 5.8 shows that the effects of PGD and MIM on the original model are comparable to each other, while FGSM is less effective than the two iterative methods.

Furthermore, adversarial training has been adopted in AdvLoc to combat adversarial attacks. Since the the verification accuracy of adversarially trained localization models (i.e., FGSM-AT, MIM-AT, and PGD-AT) are very close when not being attacked, their average verification accuracy (called "Adversarial Trained Models" in Fig. 5.8) is very close to that of the original model. Thus, it is safe to say that adversarial training does not degrade the performance of the localization model when it is not attacked. With adversarial training, the verification accuracy of each model is enhanced remarkably when under adversarial attacks. For FGSM-AT, the attacked verification accuracy (the light blue line) remains above 0.74. When $\epsilon = 1$, the attacked verification accuracy of FGSM-AT reaches 0.8. Compared with the original model, FGSM-AT achieves an improvement of 0.12 in verification accuracy when $\epsilon = 0.1$, and an improvement of 0.44 when $\epsilon = 1$. In addition, the FGSM-AT curve is more stable for the whole range of ϵ, indicating that adversarial training is an effective defense against FGSM attacks. Similarly to FGSM-AT, adversarial training also strengthens the robustness of the localization model against MIM and PGD attacks, even though the extent of the enhancements are is not as notable as that of FGSM-AT. Nevertheless, the average improvements in verification accuracy achieved by MIM-AT and PGD-AT over the original model are still both greater than 0.25.

Fig. 5.9 illustrates the verification accuracy of the localization model in the corridor environment. As in Fig. 5.8, the localization model is attacked by three methods, FGSM, MIM, and PGD. Since the corridor is a LOS dominant environment, the WiFi signals do not suffer

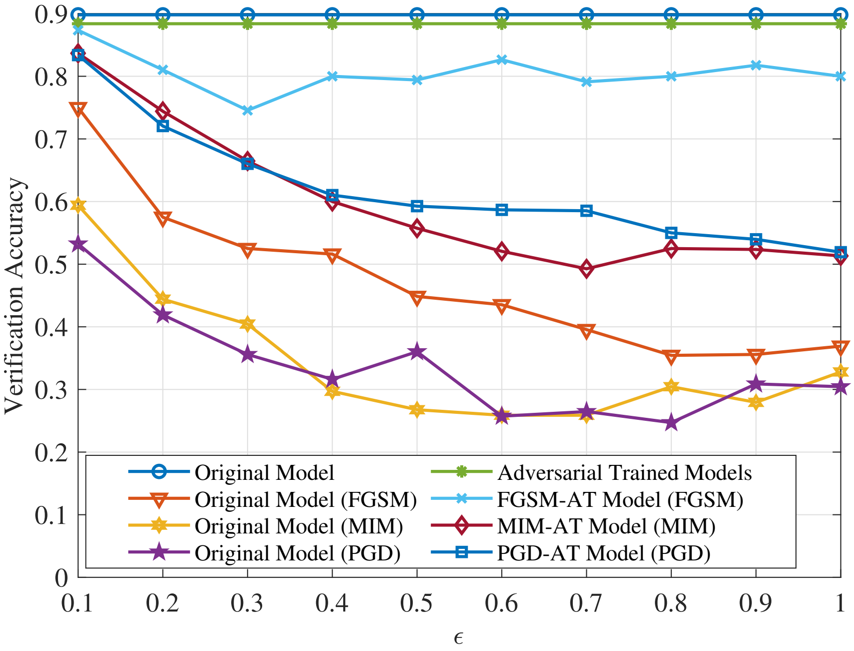


Figure 5.8: Verification accuracy of the localization models in the lab environment.

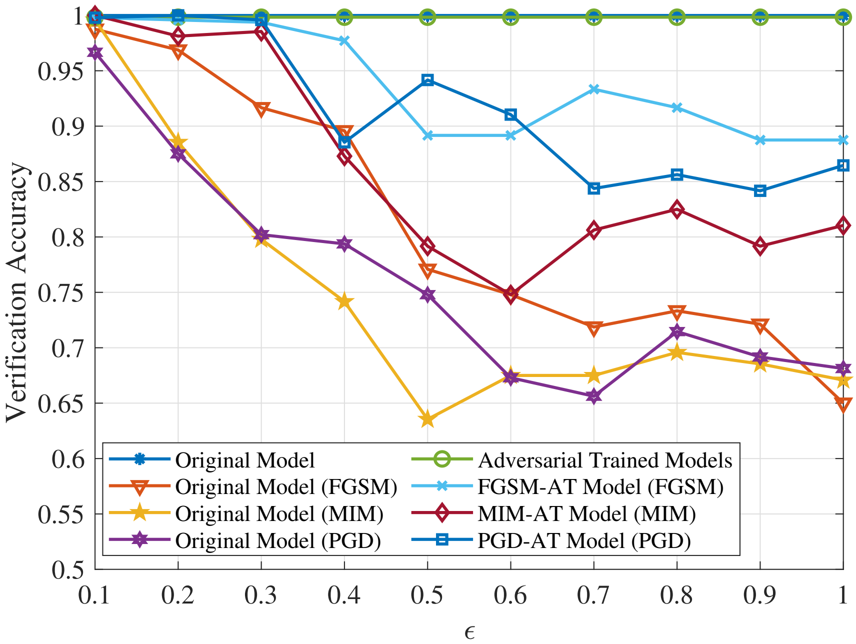


Figure 5.9: Verification accuracy of the localization models in the corridor environment.

from severe multipath effects. Therefore, the overall localization accuracy in the corridor is higher than 0.6, which is better than the lab case. With the increment of ϵ, all three attack methods contribute to degraded verification accuracy gradually, which is in accordance with the results shown inFig. 5.8. In general, PGD and MIM are more effective than FGSM, even though FGSM decreases the verification accuracy to 0.65 when $\epsilon = 1$. Moreover, adversarial

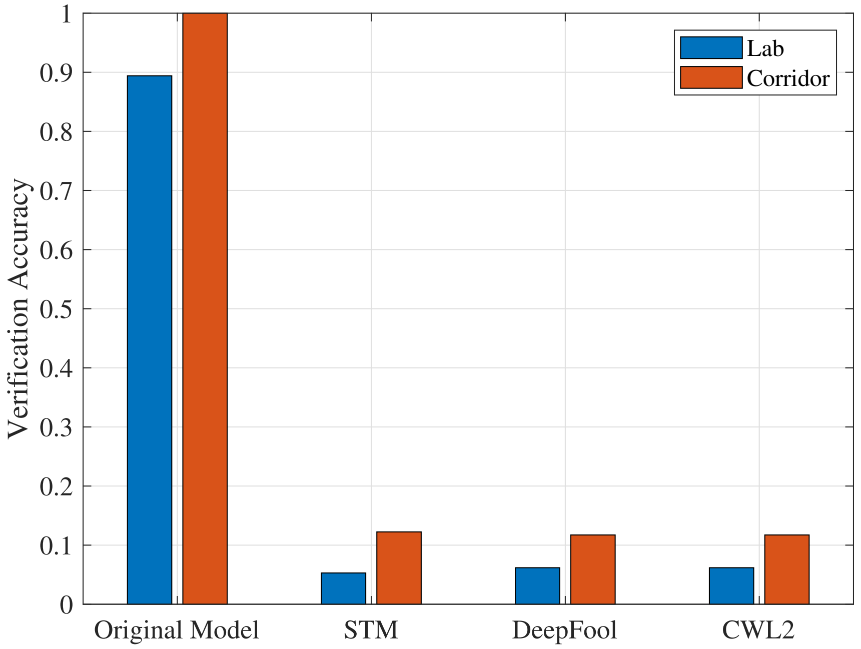


Figure 5.10: Verification accuracy of the localization models attacked by CWL2, Deepfool, and STM.

training is again an effective defense strategy for FGSM. For the adversarial examples generated by FGSM, the verification accuracy of FGSM-AT reaches 0.88 when ϵ is increased to 1. Both MIM-AT and PGD-AT also provide effective defense against the corresponding attacks, even though the extents of gains are not comparable to that of FGSM-AT.

To better evaluate the threat of adversarial attacks to indoor localization systems, three additional attack methods, STM, DeepFool, and CWL2, are also leveraged in the experiments. As shown in Fig. 5.10, the verification accuracy drops severely under these attacks. In the corridor case, all the three attacks reduce the verification accuracy to 0.13 or even worse. Similarly, the verification accuracy decreases from 0.9 to lower than 0.065 by all the attacks in the lab case. Thus, the optimization-based and the spatial transformation based attack methods are also harmful to indoor localization system. In addition, according to [176], the localization model does not acquire transferability from the adversarial training, which means the model is still vulnerable to other types of adversarial attacks even if it is trained adversarially. Thus, further investigation is needed on adversarial training to take various types of attacks into account rather than a specific attack method.

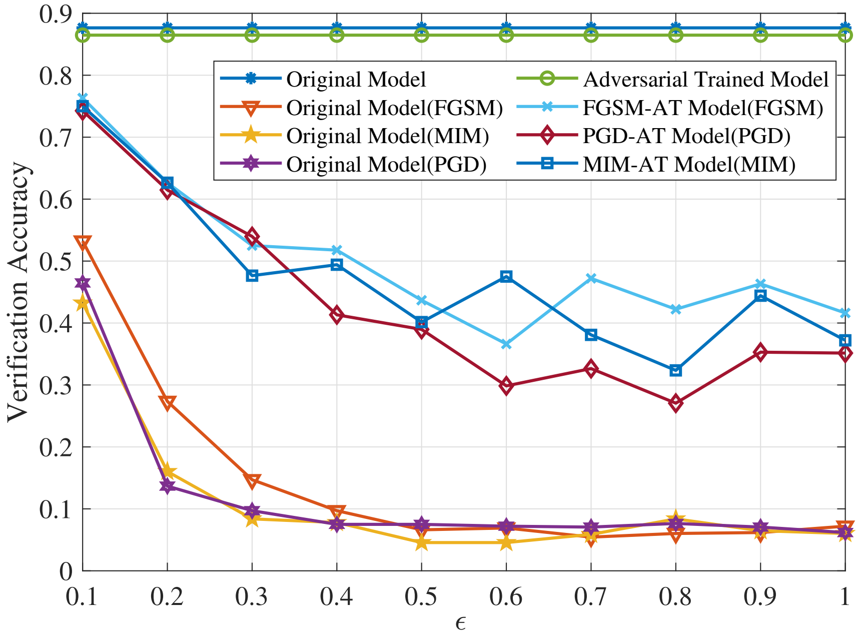


Figure 5.11: Verification accuracy of the DCNN localization models in the lab environment.

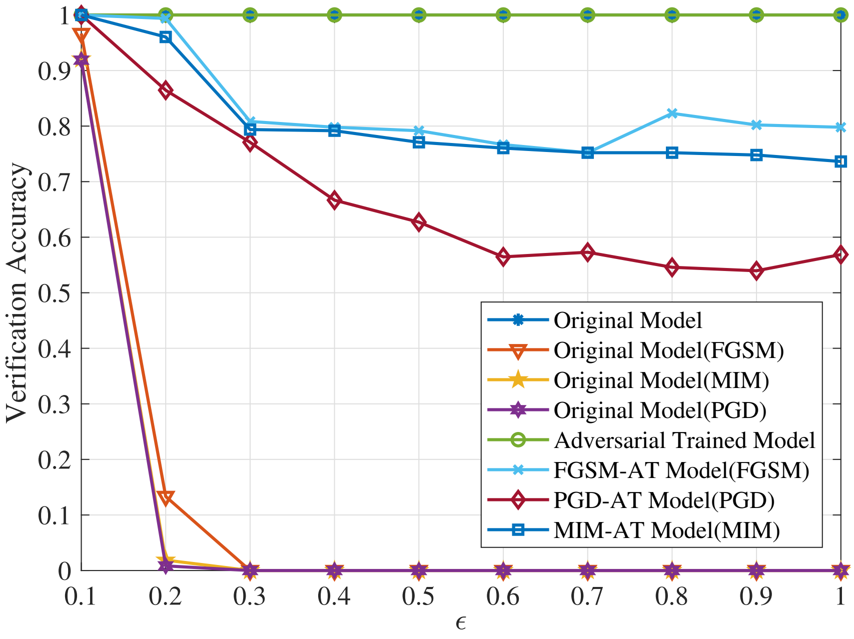


Figure 5.12: Verification accuracy of the DCNN localization models in the corridor environment.

Besides of ResNet model, we also examine the effect of adversarial attacks and adversarial training to the localization systems based on the vanilla DCNN. The network used for comparison is composed of three convolutional layers. The kernel size for each layer is 8x8, 6x6 and 5x5, respectively. 16 feature maps are generated in each convolutional layer. ReLu is leveraged as the activation function following the convolutional layers. Similar with the ResNet model, the cross-entropy loss is calculated for weight update. Fig. 5.11 and Fig. 5.12 describe the verification accuracy of the DCNN based localization model under white-box attacks in the lab and corridor environment. As is shown in Fig. 5.11, all attack methods successfully mislead the verification accuracy in the lab environment. When ϵ arrives 0.4, all verification accuracy is reduced to lower than 0.1. Because of the simpler structure of DCNN, it shows the sensitivity to the adversarial attacks. With adversarial training, the performance of all models is recovered in some extent. However, there is no clear performance difference among the models. The verification accuracy in the corridor case is displayed in Fig. 5.12. Comparing with the lab environment, the corridor case is LOS-dominant. Thus, verification accuracy of the original model keeps at 1. However, the performance breaks down as the ϵ going up to 0.2. All three attack methods reduce the verification accuracy to 0 with ϵ of 0.3. Adversarial training also reveals the effectiveness in eliminating adversarial perturbations, even though the verification accuracy is not recovered to 0.85. By examining the vanilla DCNN based localization systems, we notice that the robustness of systems is determined by the complexity and depth of the network models. The shallow networks, such as the vanilla DCNN, are completely mislead by the adversarial perturbations with a low ϵ, which hampers us to discuss the effect of the magnitude of perturbation to the system performance. Furthermore, [108] and [109] exhibits that the deeper DCNN has a better performance in the fingerprinting based indoor localization tasks. Thus, we would investigate the effect of adversarial attacks to localization system with the ResNet model.

5.4.3 Location Error Under White-box Attacks

Even though location estimation is treated as a multi-class classification problem in DCNN based localization systems, a uniqueness challenge in such localization systems is that the class of the online input to the trained model usually does not belong to an existing class in the offline training dataset. For example, we labeled the CSI data collected from the position between point-A and point-B with A in the testing dataset. The location prediction could be correct only if the localization system produces the same label. However, the testing position is actually between point-A and point-B. Obviously, it is unfair to say that the location prediction is wrong when the prediction from the system is B. To address this issue, the output of the DCNN is usually used as similarity to calculate the estimated location using a Bayesian method (see Section 5.3.1). Thus, the test accuracy in the online stage may not precisely evaluate the performance of the localization system. In this chapter, the location error is also utilized to measure the effect of adversarial attacks and adversarial training on the localization system.

First, we examine the performance of AdvLoc in the lab setting. Fig. 5.13 presents the location errors of FGSM-AT when attacked by FGSM, and of the original model when attacked by FGSM in verification and online testing. The blue dashed line is the online location error of the original model using clean inputs in online testing, while the verification location error for the same setup is denoted by a red dashed line. The errors are $2.28m$ and $0.47m$, respectively. It is obvious that the verification error rises with the increment of ϵ when the localization model is under attack, which is consistent with the verification accuracy shown in Fig. 5.8. For the online testing error, it also keeps going up along with the rise of ϵ. When $\epsilon = 0.1$, the adversarial examples increase the online testing error to $2.368m$. The highest online testing error, $2.613m$, occurs when $\epsilon = 1$. Furthermore, the performance of adversarial training is verified in Fig. 5.13 as well. Based on the FGSM-AT model, the upward trend of location errors in verification and online testing disappears. The online testing error of FGSM-AT stays around the error of the original model that leverages clean inputs. Even if $\epsilon = 1$, the increment of location error is only about $0.04m$, which is negligible in a lab environment. For the verification error, FGSM-AT guarantees that no verification error is higher than $0.81m$ when the model is under attack. It is noteworthy that the verification error declines from $2.08m$ to $0.70m$, when ϵ is fixed at 1, once adversarial training is leveraged in the localization model.

For the corridor case, the location errors of FGSM-AT attacked by FGSM and the original localization model attacked by FGSM are shown in Fig. 5.14. Compared with Fig. 5.13, the upward trend of errors in the corridor case is not as obvious as that of in the lab case. For the online testing error when the original localization model is attacked by FGSM, the error does not increase with ϵ, even though FGSM deteriorates the localization error from $1.36m$ to $1.52m$ on average. The verification location errors reveal a similar behavior. The maximum of

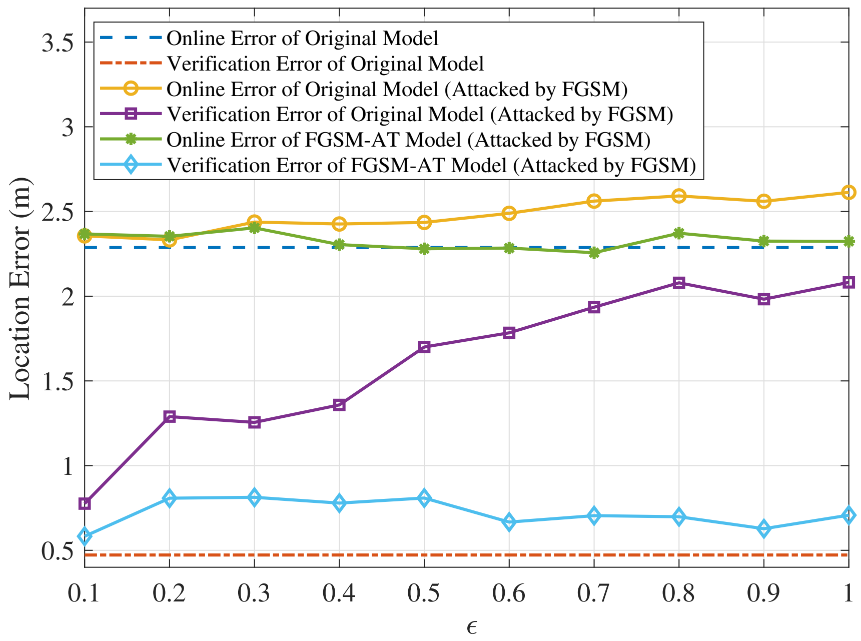


Figure 5.13: Location error of the localization models when attacked by FGSM in the lab environment.

the verification error increment is only $0.32m$ when the original localization model is attacked by FGSM with $\epsilon = 0.6$. Adversarial training is still an effective defense strategy against FGSM in the corridor case. The green line in Fig. 5.14 represents the online testing errors when FGSM-AT is attacked by FGSM. As we can see, the errors of FGSM-AT is obviously lower than that of the original model attacked by FGSM. The average error of FGSM-AT is $1.36m$, which is closed to the average error of the original model with clean inputs, i.e., $1.3504m$.

The effect of MIM and the corresponding adversarial training on location error is depicted in Fig. 5.15 and Fig. 5.16, respectively. The verification error of the original model grows significantly when attacked by MIM, which is consistent with the results presented in Fig. 5.8. Furthermore, MIM causes much larger errors than FGSM. In Fig. 5.15, the verification location error reaches $2.36m$ when attacked by adversarial examples generated by MIM with $\epsilon = 1$, which is much higher than that of FGSM. A similar phenomenon is observed in the corridor case. The verification error reaches $0.62m$ when $\epsilon = 1$, whereas the verification error is only $0.44m$ when $\epsilon = 1$ with FGSM. MIM is thus a stronger attack method than FGSM. Additionally, Fig. 5.15 shows that MIM-AT does not effectively eliminate the effect of MIM. However,

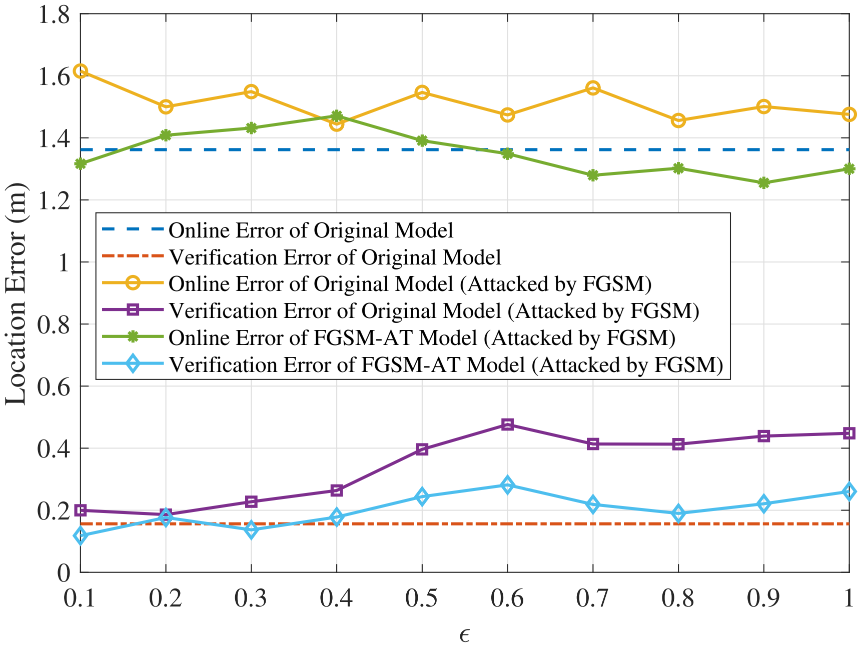


Figure 5.14: Location error of the localization model attacked by FGSM in the corridor environment.

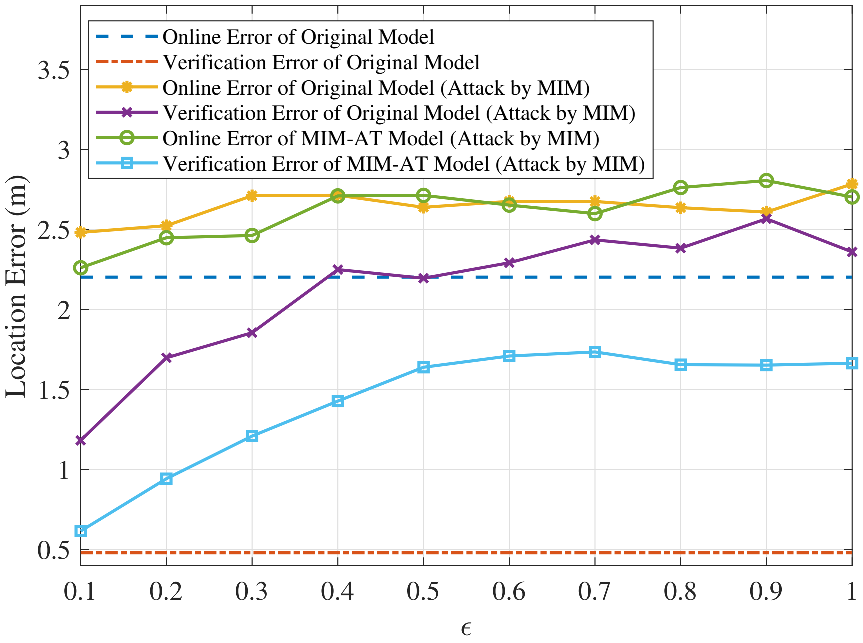


Figure 5.15: Location error of the localization models attacked by MIM in the lab environment.

adversarial training successfully removes the rising trend of the online testing error in the corridor case with MIM-AT. According to Fig. 5.16, the MIM-AT model has a commensurable performance as the unattacked original model.

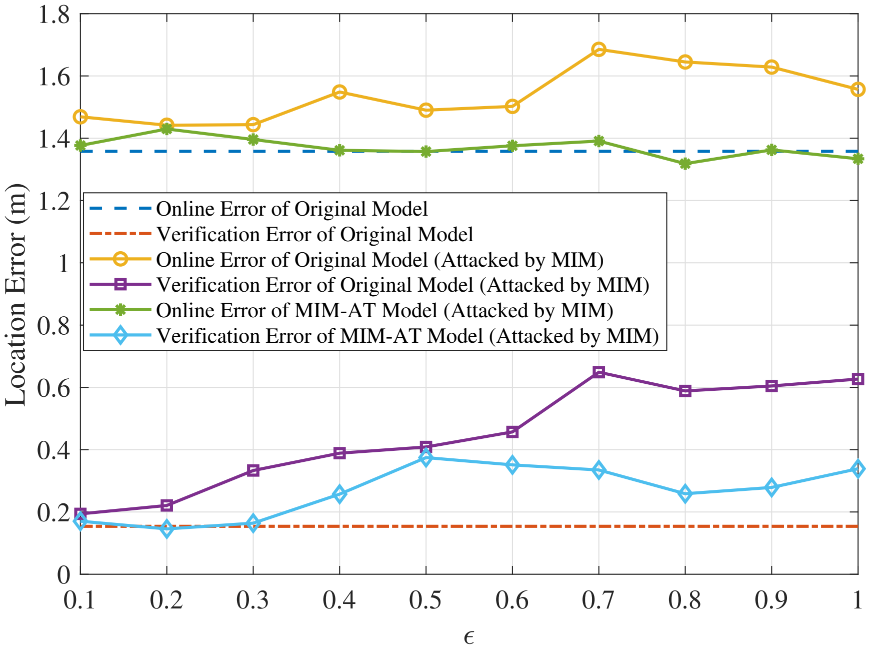


Figure 5.16: Location error of the localization models attacked by MIM in the corridor environment.

Fig. 5.17 and Fig. 5.18 present the location errors of PGD related experiments. First, the location errors in the lab case are given in Fig. 5.17. Similarly to MIM, PGD, as an iterative attack method, degrades the verification precision remarkably. The location errors climb up with the increase of ϵ when the localization model is attacked by PGD. Nevertheless, the online testing error is not improved by adversarial training in the lab case, which similar to the MIM related experiments. In the corridor case, adversarial training effectively enhances the online testing precision and verification precision.

It can be seen from Figs. 5.14, 5.16, and 5.18 that adversarial training could always reduce both online testing errors and verification errors in the corridor case. Moreover, the adversarial attacks, such as FGSM, MIM, and PGD, could not degrade much the performance of the localization model in the corridor environment. This is because the multipath effect is not as strong in the corridor case, and it is relatively easier for the DCNN model to distinguish the WiFi signals from different locations. Such "easy-to-distinguish" signals contribute to the robustness of the model, especially when the size of the training dataset is not large. As a result, the effectiveness of adversarial attacks is constrained in the corridor case, and adversarial training is also more effective. In the lab case, the received WiFi signal is a superposition of the signals

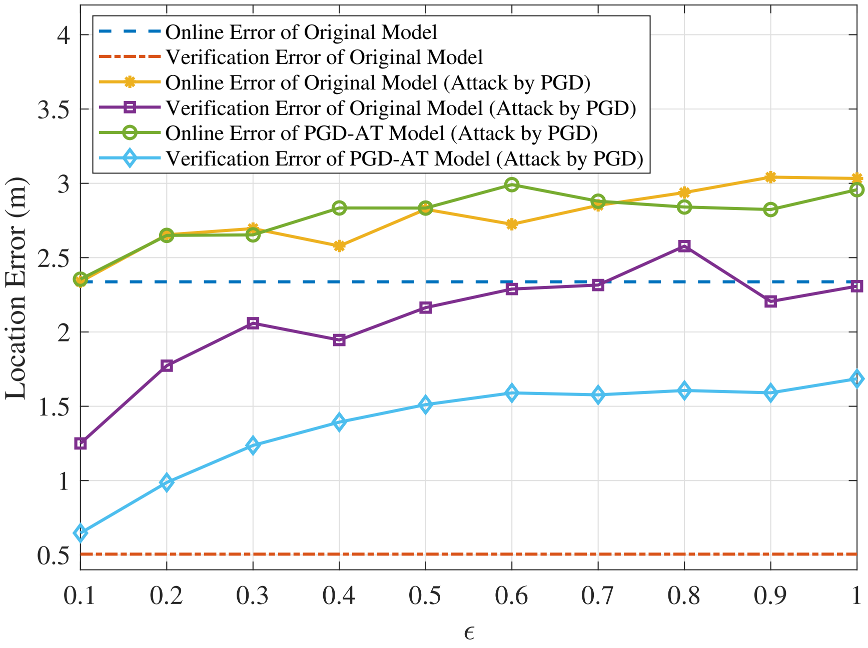


Figure 5.17: Location error of the localization models attacked by PGD in the lab environment.

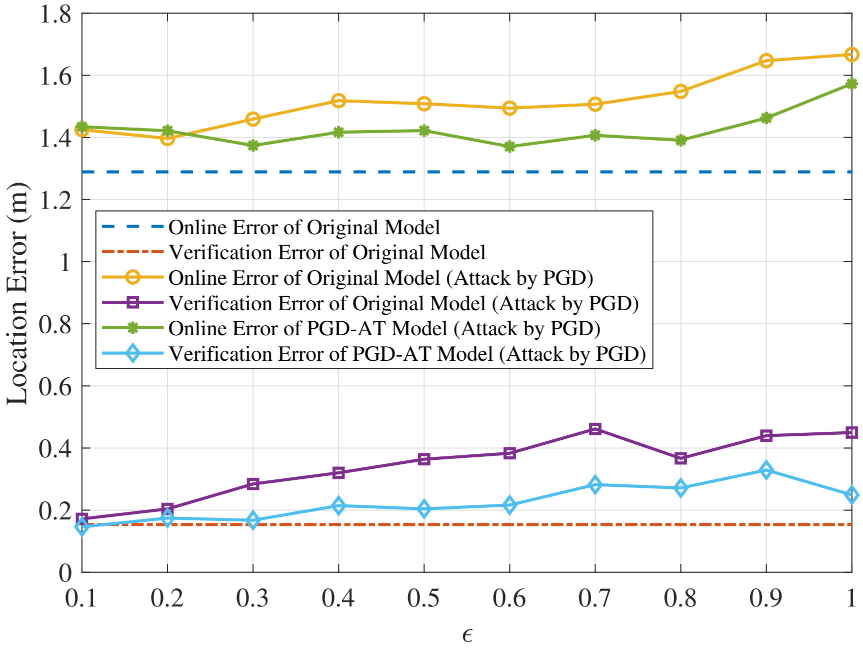


Figure 5.18: Location error of the localization models attacked by PGD in the corridor environment.

from multiple paths. The localization model becomes more gullible in facing with such noisy signals. Moreover, considering the fact that the class of the new CSI tensors in the online stage usually does not belong to any class used in offline training, such noisy signals make adversarial training struggle in the online testing. Hence, even though adversarial training achieves an

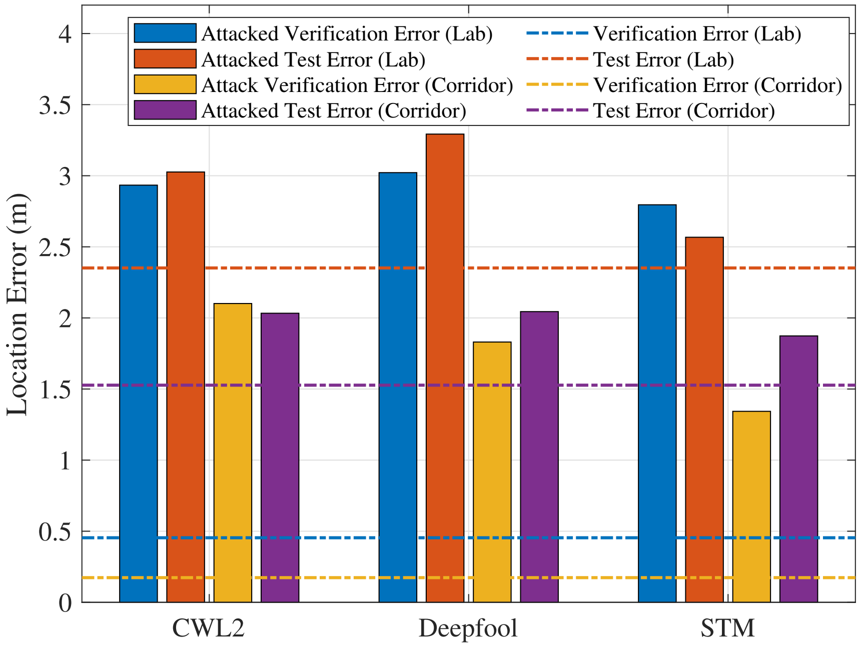


Figure 5.19: Location error of the localization models attacked by CWL2, Deepfool, and STM in the white-box scenario.

acceptable performance in defending FGSM attacks, it is not as effective for stronger attacks, such as MIM and PGD, in the online stage.

We also examine the effect of optimization based and spatial transformation based attack methods, including CWL2, DeepFool, and STM, and their location errors in the lab and corridor environments are presented in Fig. 5.19. We find the optimization based attacks, i.e., CWL2 and DeepFool, cause higher location errors in verification and online testing. Compared with FGSM, MIM, and PGD, DeepFool poses the strongest threat to localization systems in the lab case. Moreover, both CWL2 and DeepFool increase the testing errors in the corridor case to over $2m$, which is much higher than that caused by the traditional one-step or iterative attacks.

5.4.4 Location Error Under Black-box Attacks

The white-box attacks rely on knowledge of the target DCNN model, which may not be available to adversaries in many cases. Therefore, black-box attacks would be more practical in the real world. To investigate the threat of black-box attacks and evaluate the corresponding defense strategies, we implement all the previously mentioned attack methods based on the black-box attack approach.

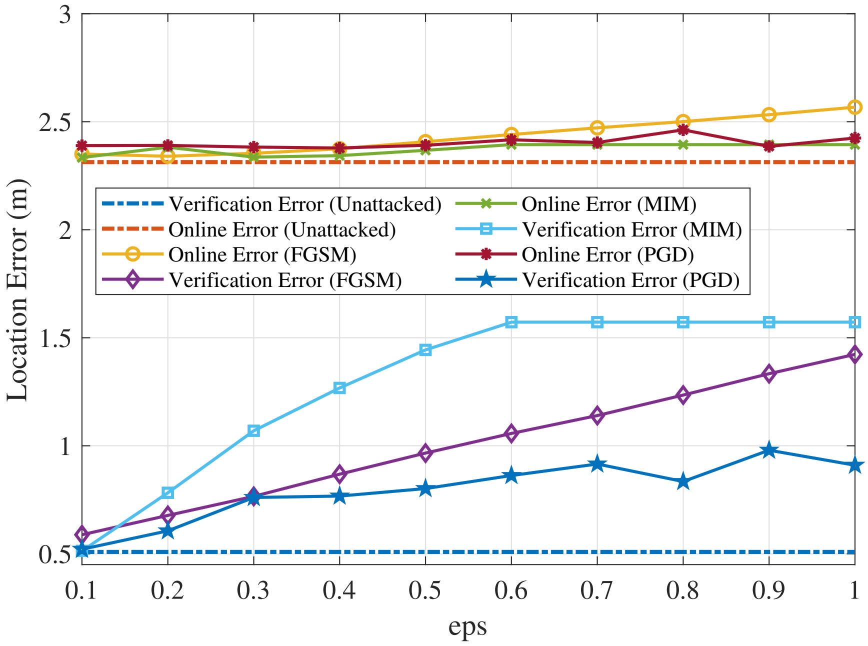


Figure 5.20: The effect of black-box attacks on the location error of the localization models in the lab environment.

First, FGSM, MIM, and PGD are deployed with the black-box approach to examine the their impacts in the lab case. As shown in Fig. 5.20, all the three attack methods exhibit outstanding performance in increasing the verification location error. However, the online testing errors are not affected by the attacks severely. The maximum increase in location error is only about $0.25m$ under FGSM generated perturbation with $\epsilon = 1$. Compared with the white-box attacks, the degradation of online testing error is negligible in Fig. 5.20.

Fig. 5.21 describes the performance of the black-box attacks in the corridor case. Because of the robustness of the localization model, the online testing errors are not influenced much by the black-box attacks. For the verification error, the maximum increment is only about $0.3m$, even though a slightly upward trend is observed in Fig. 5.21. Thus, it is safe to say that our localization model for the corridor case is robust enough against black-box attacks. In other words, the adversarial examples generated by the substitute model (i.e., ResNet-50) for black-box attack fail to mislead the original DCNN model.

We also leverage the optimization based and spacial transformation based attack methods to evaluate the system under black-box attacks. Comparing Fig. 5.19 with Fig. 5.22, we notice

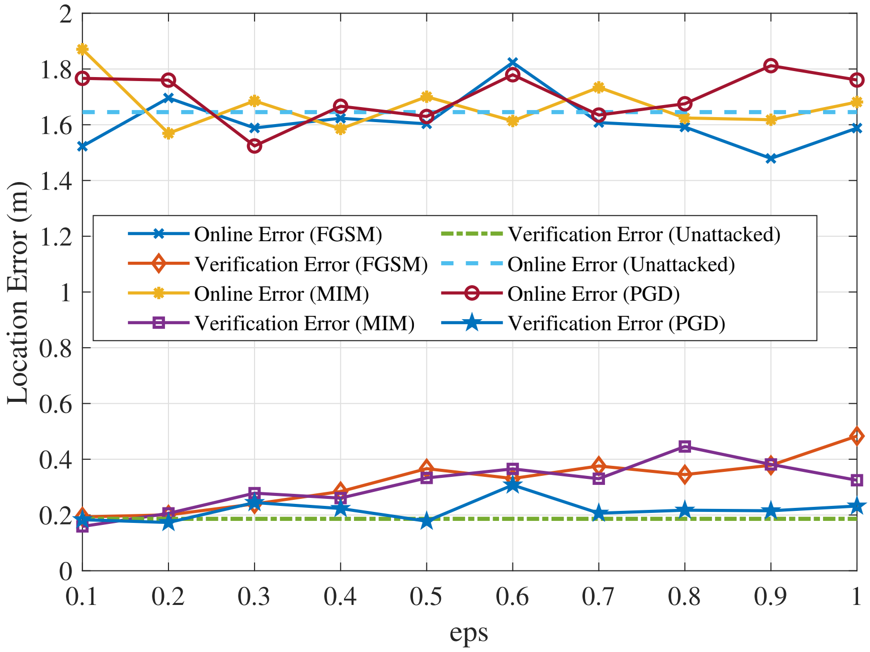


Figure 5.21: The effect of black-box attacks on the location error of the localization models in the corridor environment.

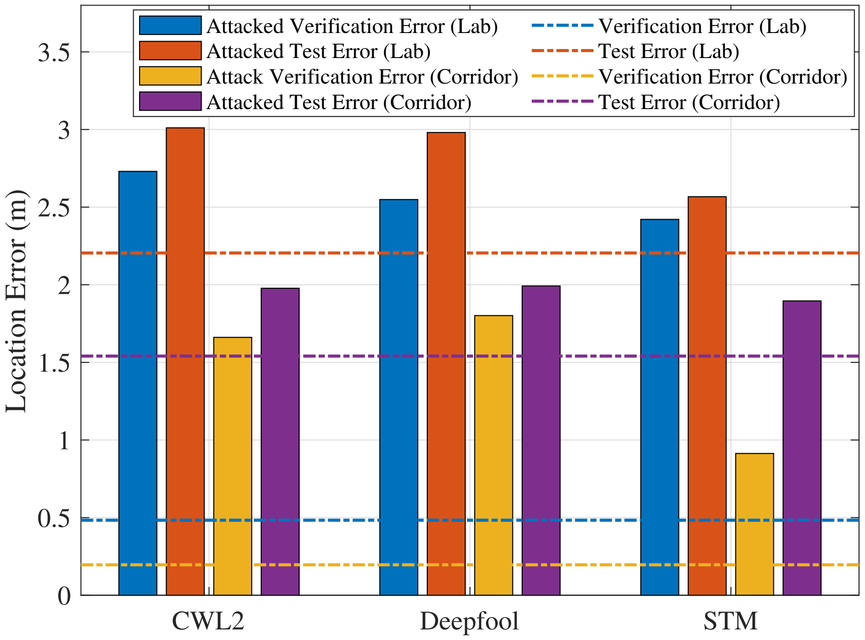


Figure 5.22: Location error of the localization models attacked by CWL2, DeepFool, and STM in the black-box scenario.

that each result in Fig. 5.22 is lower than the corresponding result in Fig. 5.19. CWL2, DeepFool, and STM could not achieve similar performance when used for black-box attack. The difference in the knowledge between the black-box model (i.e., ResNet-18) and the substitute model (i.e., ResNet-50) limits the performance of the attacks.

5.5 Conclusions

In this chapter, we presented AdvLoc, an Adversarial Deep Learning for Indoor Localization system using CSI Tensors, which is resilient against the typical first-order adversarial attacks. With the proposed AdvLoc system, we analyzed the effect of six types common adversarial attacks in both white-box attack and black-box attack scenarios. The extensive experimental study exposed the threat of the adversarial attacks to indoor localization systems and validated the superior performance of the proposed AdvLoc system in defending against first-order adversarial attacks.

Chapter 6

Summary and Future Work

6.1 Summary

During my Ph.D. program, I focused on developing practical methods to alleviate the intrinsic problems related to fingerprinting-based indoor localization. The proposed systems focus not just on WiFi, but also on RFID systems. To begin, DeepMap uses RSS to generate precise indoor radio maps. The nonlinear properties are reproduced using DGP to create the radio map with a limited number of signal fingerprints. With the DGP model, we studied the uncertainty information to optimize the localization accuracy in MapLoc. MapLoc leverages the bimodal inputs, WiFI RSS and magnetic field intensity, to build the fingerprint dataset. The uncertainty information describes the reliability of signal measurements in the fingerprint. Moreover, the sequence information related to the physical constraints of the indoor environment and the motion model is also leveraged in the network training of the MapLoc system. In the MulTLoc system, phase information is leveraged to improve the localization accuracy for achieving centimeter-level location estimation. The hologram tensors are used to replace traditional fingerprints, such as AoA images, to advance the transferability of the system. With the hologram tensors, the ambiguity between fingerprints is preserved. Thus, the classification problem could be treated as a regression problem in the MulTLoc system. And the network design is decoupled from the hardware configuration, which enhances the compatibility of the MulTLoc system. Besides of previous systems, AdvLoc is proposed to study the threat of adversarial attacks to the fingerprinting based indoor localization systems using deep neural networks. Six types of popular adversarial attacks are implemented in the system using both white-box and black-box methods. The experimental results show that adversarial attacks pose a threat to deep learning-based indoor localization systems. In addition, adversarial training is used in the system to increase the robustness of the localization system.

6.2 Future Work

Even though some preliminary attempts are conducted in this dissertation to alleviate the intrinsic issues related to fingerprint methods, many interesting problems are still open in the field of indoor localization.

6.2.1 Dynamic Environment

The dynamic environment is always a challenge for the fingerprinting-based indoor localization system. Both RSS and phase information are vulnerable to temporal and spatial variance. To guarantee the system accuracy, an effective method of fingerprint update has to be investigated to enhance the robustness of the localization system. On the other hand, the dynamic environment also represents the indoor scenarios including multiple moving objects. The moving objects would generate unpredictable multipath, which hampers the phase localization significantly. Performing additional research on self-supervised learning could be a viable method for extracting static features from a signal acquired in a dynamic context.

6.2.2 Sensor Fusion and Multimodal Data

With the proliferation of IoT devices, it is typical for multiple types of sensors to be equipped with the end devices. Therefore, sensor fusion would be a effective tools to bring together inputs from multiple sensors, such as WiFi, RFID, FMCW radar, camera, microphone, IMU, and magnetometers. Even though the associated multimodal data would provide some difficulties for data processing, the opportunity to improve the accuracy and resilience of wireless sensing systems would be enticing.

6.2.3 Dataset for indoor localization

In this dissertation, several deep neural networks are deployed to complete the classification and regression tasks. However, all the networks were evaluated with a homebrew dataset. Although some public WiFi or RFID datasets are available online, the scale of the dataset is usually limited. As a result, the density of the fingerprints would be a constraint for the localization accuracy. Moreover, the signal quality is another drawback for the current open dataset. Most datasets only provide a signal sample at a specific time, which could not represent the temporal and spatial variance of wireless signals. Thus, we hope to build a public indoor localization dataset that covers a large area with dense fingerprints. The signal would include but not be limited to WiFi, RFID, and FMCW radar. And the data would be collected in more than one time slot. It would be an ideal benchmark for researchers to evaluate their related works.

List of Publications

- Xiangyu Wang, Xuyu Wang, Shiwen Mao, Jian Zhang, Senthilkumar C.G. Periaswamy, and Justin Patton, "Adversarial deep learning for indoor localization," IEEE Internet of Things Journal, to appear. DOI: 10.1109/JIOT.2022.3155562.
- Xiangyu Wang, Xuyu Wang, and Shiwen Mao, "Indoor fingerprinting with bimodal CSI tensors: A deep residual sharing learning approach," IEEE Internet of Things Journal, vol.8, no.6, pp.4498-4513, Mar. 2021.
- Xiangyu Wang, Xuyu Wang, Shiwen Mao, Jian Zhang, Senthilkumar CG Periaswamy, and Justin Patton, "Indoor radio map construction and localization with deep Gaussian Processes," IEEE Internet of Things Journal, vol.7, no.11, pp. 11238-11249, Nov. 2020.
- Xiangyu Wang, Jian Zhang, Zhitao Yu, Shiwen Mao, Senthilkumar C.G. Periaswamy, and Justin Patton, "On remote temperature sensing using commercial UHF RFID tags," IEEE Internet of Things Journal, vol.6, no.6, pp. 10715-10727, Dec. 2019.
- Jian Zhang, Xiangyu Wang, Zhitao Yu, Yibo Lyu, Shiwen Mao, Senthilkumar CG Periaswamy, Justin Patton, and Xuyu Wang*, "Robust RFID based 6-DoF localization for unmanned aerial vehicles," IEEE Access Journal, Special Section on Network Resource Management in Flying Ad Hoc Networks: Challenges, Potentials, Future Applications, and Wayforward, vol.7, no.1, pp. 77348-77361, June 2019.
- Jian Zhang, Zhitao Yu, Xiangyu Wang, Yibo Lyu, Shiwen Mao, Senthilkumar CG Periaswamy, Justin Patton, and Xuyu Wang*, "RFHUI: An RFID based human-unmanned aerial vehicle interaction system in an indoor environment," Elsevier Digital Communications and Networks Journal, vol.6, no.1, pp.14-22, Feb. 2020.

- Xuyu Wang, Xiangyu Wang, and Shiwen Mao, “RF sensing for Internet of Things: A general deep learning framework,” IEEE Communications, Feature Topic on Exploring Caching, Communications, Computing and Security for the Emerging Smart Internet of Things, vol.56, no.9, pp.62-67, Sept. 2018.

- Xuyu Wang, Xiangyu Wang, and Shiwen Mao, “Deep convolutional neural networks for indoor localization with CSI images,” IEEE Transactions on Network Science and Engineering, Special Issue on Network Science for Internet of Things (IoT), vol.7, no.1, pp.316-327, Jan./Mar. 2020.

- Xiangyu Wang, Jian Zhang, Shiwen Mao, Senthilkumar C.G. Periaswamy, and Justin Patton, “Locating multiple RFID tags with Swin Transformer-based RF hologram tensor filtering,” in Proc. IEEE VTC-Fall 2022, London, UK, Sept. 2022.

- Xiangyu Wang, Jian Zhang, Shiwen Mao, Senthilkumar C.G. Periaswamy, and Justin Patton, “MulTLoc: RF hologram tensor filtering and upscaling for indoor localization using multiple UHF passive RFID tags,” invited paper, in Proc. ICCCN 2021, Athens, Greece, July 2021.

- Xiangyu Wang, Jian Zhang, Zhitao Yu, Eric Mao, Senthilkumar C. G. Periaswamy and Justin Patton, ”RFThermometer: A Temperature Estimation System with Commercial UHF RFID Tags,” in Proc. IEEE ICC 2019, Shanghai, China, May, 2019, pp.1-6.

- Xiangyu Wang, Xuyu Wang, Shiwen Mao, Jian Zhang, Senthilkumar C.G. Periaswamy, and Justin Patton, “DeepMap: Deep Gaussian Process for indoor radio map construction and location estimation,” in Proc. IEEE GLOBECOM 2018, Abu Dhabi, United Arab Emirates, Dec. 2018, pp.1-7.

- Mohini Patil, Xuyu Wang, Xiangyu Wang, and Shiwen Mao, “Adversarial attacks on deep learning-based floor classification and indoor localization,” invited paper, in Proc. 2021 ACM Workshop on Wireless Security and Machine Learning (WiseML’21), Abu Dhabi, United Arab Emirates, June-July 2021, pp.7-12.

- Jian Zhang, Zhitao Yu, Xiangyu Wang, Yibo Lyu, Shiwen Mao, Senthilkumar C.G. Periaswamy, Justin Patton, and Xuyu Wang*, “RFHUI: An intuitive and easy-to-operate human-UAV interaction system for controlling a UAV in a 3D space,” in Proc. EAI MobiQuitous 2018, New York City, NY, Nov. 2018, pp.69-76.

- Xuyu Wang, Xiangyu Wang, and Shiwen Mao, “ResLoc: Deep residual sharing learning for indoor localization with CSI tensors,” in Proc. IEEE PIMRC 2017, Montreal, QC, Canada, Oct. 2017.

- Xuyu Wang, Xiangyu Wang, and Shiwen Mao, “CiFi: Deep convolutional neural networks for indoor localization with 5GHz Wi-Fi,” in Proc. IEEE ICC 2017, Paris, France, May 2017, pp.1-6.

References

[1] T. Zhang and S. Mao, “Machine learning for end-to-end congestion control,” *IEEE Communications Magazine*, vol. 58, no. 6, pp. 52–57, June 2020.

[2] M. Feng, S. Mao, and T. Jiang, “Dealing with link blockage in mmwave networks: A combination of d2d relaying, multi-beam reflection, and handover,” *IEEE Transactions on Wireless Communications*.

[3] T. Zhang and S. Mao, “Energy-efficient federated learning with intelligent reflecting surface,” *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 2, pp. 845–858, June 2022.

[4] Y. Wang, T. Zhang, S. Mao, and T. Rappaport, “Directional neighbor discovery in mmWave wireless networks,” *Elsevier/KeAi Digital Communications and Networks Journal*, vol. 7, no. 1, pp. 1–16, Feb. 2021.

[5] K. Xiao, S. Mao, and J. Tugnait, “TCP-Drinc: Smart congestion control based on deep reinforcement learning,” *IEEE Access Journal*, vol. 7, no. 1, pp. 11 892–11 904, Jan. 2019.

[6] X. Wang, C. Yang, and S. Mao, “On csi-based vital sign monitoring using commodity wifi,” *ACM Trans. Comput. Healthcare*, vol. 1, no. 3, may 2020. [Online]. Available: https://doi.org/10.1145/3377165

[7] X. Wang, R. Huang, and S. Mao, “Sonarbeat: Sonar phase for breathing beat monitoring with smartphones,” in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, 2017, pp. 1–8.

[8] J. Zhang, X. Wang, Z. Yu, Y. Lyu, S. Mao, S. C. Periaswamy, J. Patton, and X. Wang, "Robust rfid based 6-dof localization for unmanned aerial vehicles," *IEEE Access*, vol. 7, pp. 77 348–77 361, 2019.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR'16*, Las Vegas, NV, June 2016, pp. 770–778.

[10] S. Mao, Y. T. Hou, X. Cheng, H. D. Sherali, S. F. Midkiff, and Y.-Q. Zhang, "On routing for multiple description video over wireless ad hoc networks," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 1063–1074, Oct. 2006.

[11] M. Feng, S. Mao, and T. Jiang, "Joint frame design, resource allocation and user association for massive mimo heterogeneous networks with wireless backhaul," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1937–1950, Mar. 2018.

[12] M. Feng and S. Mao, "Dealing with limited backhaul capacity in millimeter wave systems: A deep reinforcement learning approach," *IEEE Communications*, vol. 57, no. 3, pp. 50–55, Mar. 2019.

[13] Z. He, S. Mao, S. Kompella, and A. Swami, "On link scheduling in dual-hop 60 ghz mmwave networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11 180–11 192, Dec. 2017.

[14] K. Xiao, S. Mao, and J. Tugnait, "MAQ: A multiple model predictive congestion control scheme for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2614–2626, Apr. 2017.

[15] J. Zhang, Y. Lyu, J. Patton, S. C. Periaswamy, and T. Roppel, "BFVP: A probabilistic UHF RFID tag localization algorithm using Bayesian filter and a variable power RFID model," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8250–8259, Oct. 2018.

[16] X. Wang, Z. Yu, and S. Mao, “Deepml: Deep lstm for indoor localization with smartphone magnetic and light sensors,” in *Proc. ICC’18)*, Kansas City, MO, July 2018, pp. 1–6.

[17] C. Yang, X. Wang, and S. Mao, “SparseTag: High-precision backscatter indoor localization with sparse RFID tag arrays,” in *Proc. IEEE SECON 2019*, Boston, MA, June 2019, pp. 1–9.

[18] H. Zhu, Y. Zhang, Z. Liu, S. Chang, and Y. Chen, “Hyperear: Indoor remote object finding with a single phone,” in *Proc. IEEE ICDCS’19*, Dallas, Texas, July 2019, pp. 678–687.

[19] L. Yang, Y. Chen, X. Li, C. Xiao, M. Li, and Y. Liu, “Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices,” in *Proc. ACM Mobicom’14*. ACM, Sept. 2014, pp. 237–248.

[20] X. Wang, S. Mao, S. Pandey, and P. Agrawal, “CA2T: Cooperative antenna arrays technique for pinpoint indoor localization,” in *Proc. MobiSPC 2014*, Niagara Falls, Canada, Aug. 2014, pp. 392–399.

[21] X. Wang, H. Zhou, S. Mao, S. Pandey, P. Agrawal, and D. Bevly, “Mobility improves LMI-based cooperative indoor localization,” in *Proc. IEEE WCNC 2015*, New Orlean, LA, Mar. 2015, pp. 2215–2220.

[22] J. Zhang, Z. Yu, X. Wang, Y. Lyu, S. Mao, S. C. Periaswamy, J. Patton, and X. Wang, “Rfhui: An intuitive and easy-to-operate human-uav interaction system for controlling a uav in a 3d space,” in *Proc. MobiQuitous ’18*, New York, NY, Nov. 2018, pp. 69—-76.

[23] Y. Xie, J. Xiong, M. Li, and K. Jamieson, “md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking,” in *Proc. ACM Mobocom’19*, Los Cabos, Mexico, Oct. 2019, pp. 1–16.

[24] H. Liu, H. Darabi, P. Banerjee, and L. Jing, “Survey of wireless indoor positioning techniques and systems,” *IEEE Trans. Syst., Man, Cybern. C*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[25] P. Bahl and V. N. Padmanabhan, “Radar: An in-building RF-based user location and tracking system,” in *Proc. IEEE INFOCOM’00*, Tel Aviv, Israel, Mar. 2000, pp. 775–784.

[26] M. Youssef and A. Agrawala, “The Horus WLAN location determination system,” in *Proc. ACM MobiSys’05*, Seattle, WA, June 2005, pp. 205–218.

[27] 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, Jan. 2017.

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

[29] X. Wang, L. Gao, S. Mao, and S. Pandey, “BiLoc: Bi-modal deep learning for indoor localization with commodity 5GHz WiFi,” *IEEE Access J.*, vol. 5, pp. 4209–4220, Mar. 2017.

[30] X. Wang, X. Wang, and S. Mao, “Cifi: Deep convolutional neural networks for indoor localization with 5 ghz wi-fi,” in *Proc. IEEE ICC 2017*, Paris, France, May 2017, pp. 1–6.

[31] ——, “ResLoc: Deep residual sharing learning for indoor localization with CSI tensors,” in *Proc. IEEE PIMRC 2017*, Montreal, Canada, Oct. 2017, pp. 1–6.

[32] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, “GPPS: A Gaussian process positioning system for cellular networks,” in *Proc. NIPS’03*, Whistler, BC, Canada, Dec. 2003, pp. 579–586.

[33] S. He and S.-H. G. Chan, “Towards crowdsourced signal map construction via implicit interaction of iot devices,” in *Proc. IEEE SECON’17*, San Diego, CA, June 2017, pp. 1–9.

[34] F. Duvallet and A. D. Tews, “Wifi position estimation in industrial environments using gaussian processes,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, Sept. 2008, pp. 2216–2221.

[35] M. Dashti, S. Yiu, S. Yousefi, F. Perez-Cruz, and H. Claussen, “Rssi localization with gaussian processes and tracking,” in *Proc. IEEE Globecom’15 Workshops*, San Diego, CA, Dec. 2015, pp. 1–6.

[36] W. Zhang, H. Huang, , and X. Tian, “Gaussian process based radio map construction for LTE localization,” in *Proc. IEEE WCSP’17*, Nanjing, China, Oct. 2017, pp. 1–6.

[37] F. Seco, C. Plagemann, A. R. Jiménez, and W.Burgard, “Improving RFID-based indoor positioning accuracy using Gaussian processes,” in *Proc. IEEE IPIN’10*, ETH Zurich, Sept. 2010, pp. 1–8.

[38] H. Wymeersch, S. Maranò, W. M. Gifford, and M. Z. Win, “A machine learning approach to ranging error mitigation for UWB localization,” *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1719–1728, June 2012.

[39] C. E. Rasmussen and C. K. I. Williams, “Gaussian processes in machine learning,” in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[40] A. Damianou and N. Lawrence, “Deep Gaussian processes,” in *Proc. 16th Int. Conf. Artificial Intell. Stat.*, Scottsdale, AZ, May 2013, pp. 207–215.

[41] X. Wang, L. Gao, and S. Mao, “PhaseFi: Phase fingerprinting for indoor localization with a deep learning approach,” in *Proc. GLOBECOM’15*, San Diego, CA, Dec. 2015, pp. 1–6.

[42] M. Titsias and N. D. Lawrence, “Bayesian gaussian process latent variable model,” in *Proc. 13th Int. Conf. Artificial Intell. Stat.*, Sardinia, Italy, May 2010, pp. 844–851.

[43] S. Kumar, R. M. Hegde, and N. Trigoni, “Gaussian process regression for fingerprinting based localization,” *Elsevier Ad Hoc Netw.*, vol. 51, pp. 1–10, Nov. 2016.

[44] G. Jekabsons and V. Zuravlyov, “Refining wi-fi based indoor positioning,” in *Proceedings of 4th International Scientific Conference Applied Information and Communication Technologies (AICT)*, Jelgava, Latvia, 2010, pp. 87–95.

[45] J. Xiao, K. Wu., Y. Yi, and L. Ni, “FIFS: Fine-grained indoor fingerprinting system,” in *Proc. IEEE ICCCN’12*, Munich, Germany, Aug. 2012, pp. 1–7.

[46] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, “You are facing the mona lisa: Spot localization using phy layer information,” in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 183–196.

[47] X. Wang, L. Gao, S. Mao, and S. Pandey, “DeepFi: Deep learning for indoor fingerprinting using channel state information,” in *Proc. WCNC’15*, New Orleans, LA, Mar. 2015, pp. 1666–1671.

[48] J. Wang, X. Zhang, Q. Gao, H. Yue, and H. Wang, “Device-free wireless localization and activity recognition: A deep learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6258–6267, 2016.

[49] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, and M. Youssef, “WiDeep: WiFi-based accurate and robust indoor localization system using deep learning,” in *IEEE PerCom’19*, Kyoto, Japan, Mar. 2019, pp. 1–10.

[50] X. Chen, C. Ma, M. Allegue, and X. Liu, “Taming the inconsistency of Wi-Fi fingerprints for device-free passive indoor localization,” in *Proc. IEEE INFOCOM’ 2017*, Atlanta, GA, May 2017, pp. 1–9.

[51] J. Wang, X. Zhang, Q. Gao, H. Yue, and H. Wang, “Device-free wireless localization and activity recognition: A deep learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6258–6267, July 2017.

[52] W. Shao, H. Luo, F. Zhao, Y. Ma, Z. Zhao, and A. Crivello, “Indoor positioning based on fingerprint-image and deep learning,” *IEEE Access*, vol. 6, pp. 74 699–74 712, 2018.

[53] H. Zou, M. Jin, H. Jiang, L. Xie, and C. Spanos, “Winips: Wifi-based non-intrusive indoor positioning system with online radio map construction and adaptation,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8118–8130, 2017.

[54] Y. Tao and L. Zhao, “A novel system for wifi radio map automatic adaptation and indoor positioning,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 683–10 692, 2018.

[55] B. Jia, B. Huang, H. Gao, W. Li, and L. Hao, “Selecting critical wifi aps for indoor localization based on a theoretical error analysis,” *IEEE Access*, vol. 7, pp. 36 312–36 321, Mar. 2019.

[56] P. Huang, H. Zhao, W. Liu, and D. Jiang, “Maps: indoor localization algorithm based on multiple ap selection,” *Mobile Networks and Applications*, vol. 26, no. 2, pp. 649–656, Apr. 2021.

[57] X. Shi, J. Guo, and Z. Fei, “Wlan fingerprint localization with stable access point selection and deep lstm,” in *Proc. IEEE ICICN’20*, Xi’an, China, Aug. 2020, pp. 56–62.

[58] X. Wang, Z. Yu, and S. Mao, “Indoor localization using magnetic and light sensors with smartphones: A deep LSTM approach,” *Springer Mobile Networks and Applications (MONET) Journal*, vol. 25, no. 2, pp. 819–832, Apr. 2020.

[59] A. H. Salamah, M. Tamazin, M. A. Sharkas, and M. Khedr, “An enhanced WiFi indoor localization system based on machine learning,” in *Proc. IEEE IPIN’16*, Alcala de Henares, Spain, Oct. 2016, pp. 1–8.

[60] E. Jedari, Z. Wu, R. Rashidzadeh, and M. Saif, “Wi-Fi based indoor location positioning employing random forest classifier,” in *Proc. IEEE IPIN’15*, Banff, Canada, Oct. 2015, pp. 1–5.

[61] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional neural networks based indoor wi-fi localization using channel state information," *IEEE Access*, vol. 5, no. 1, pp. 18 066–180 747, Sept. 2017.

[62] X. Wang, X. Wang, S. Mao, J. Zhang, S. Periaswamy, and J. Patton, "Indoor radio map construction and localization with deep Gaussian Processes," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11 238–11 249, Nov. 2020.

[63] Y. Li, H. Ma, L. Wang, and S. Mao, "Optimized content caching and user association for edge computing in densely deployed heterogeneous networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 2130–2142, June 2022.

[64] M. Chen, V. C. M. Leung, S. Mao, and M. Li, "Energy-efficient itinerary planning for mobile agents in wireless sensor networks," in *Proc. IEEE ICC 2009*, Dresden, Germany, June 2009, pp. 1–5.

[65] M. Feng, S. Mao, and T. Jiang, "Boost: Base station on-off switching strategy for energy efficient massive mimo hetnets," in *Proc. IEEE INFOCOM 2016*, San Francisco, CA, Apr. 2016, pp. 1395–1403.

[66] X. Wang, S. Mao, and M. Gong, "A survey of lte wi-fi coexistence in unlicensed bands," *ACM GetMobile: Mobile Computing and Communications Review*, vol. 20, no. 3, pp. 17–23, July 2016.

[67] K. Xiao, S. Mao, and J. Tugnait, "Hierarchical radio resource allocation for network slicing in fog radio access networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3866–3881, Apr. 2019.

[68] M. Feng, S. Mao, and T. Jiang, "Joint duplex mode selection, channel allocation, and power control for full-duplex cognitive femtocell networks," *Elsevier Digital Communications and Networks Journal*, vol. 1, no. 1, pp. 30–44, Feb. 2015.

[69] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. M. Ni, "Fila: Fine-grained indoor localization," in *Proc. IEEE Infocom'12*, Orlando, FL, Apr. 2012, pp. 2210 – 2218.

[70] A. Chekuri and M. Won, "Automating WiFi fingerprinting based on nano-scale unmanned aerial vehicles," in *Proc. IEEE VTC'17-Spring*, Sydney, Australia, June 2017, pp. 1–5.

[71] S. Piao, Z. Ba, L. Su, D. Koutsonikolas, S. Li, and K. Ren, "Automating CSI measurement with UAVs: from problem formulation to energy-optimal solution," in *Proc. IEEE INFOCOM'19*, Paris, France, Apr./May 2019, pp. 2404–2412.

[72] Z. Wang, L. Liu, J. Xu, and L. Zeng, "A UAV path and action planning algorithm for indoor localization information collection," in *Proc. IEEE BDCloud'20*, Exeter, UK, Oct. 2020, pp. 609–616.

[73] H. Zou, C.-L. Chen, M. Li, J. Yang, Y. Zhou, L. Xie, and C. J. Spanos, "Adversarial learning-enabled automatic WiFi indoor radio map construction and adaptation with mobile robot," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6946–6954, Aug. 2020.

[74] C. Wu, Z. Yang, and C. Xiao, "Automatic radio map adaptation for indoor localization using smartphones," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 517–528, Mar. 2017.

[75] B. Huang, Z. Xu, B. Jia, and G. Mao, "An online radio map update scheme for WiFi fingerprint-based localization," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6909–6918, Aug. 2019.

[76] Z. Gu, Z. Chen, Y. Zhang, Y. Zhu, M. Lu, and A. Chen, "Reducing fingerprint collection for indoor localization," *Computer Communications*, vol. 83, pp. 56–63, June 2016.

[77] J. Luo and L. Fu, "A smartphone indoor localization algorithm based on WLAN location fingerprinting with feature extraction and clustering," *MDPI Sensors*, vol. 17, no. 6, p. 1339, June 2017.

[78] P. Yazdanian and V. Pourahmadi, "DeepPos: Deep supervised autoencoder network for CSI based indoor localization," *arXiv preprint arXiv:1811.12182*, Nov. 2018. [Online]. Available: https://arxiv.org/abs/1811.12182

[79] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, “Recurrent neural networks for accurate rssi indoor localization,” *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 639–10 651, Dec. 2019.

[80] Z. Chen, H. Zou, J. Yang, H. Jiang, and L. Xie, “WiFi fingerprinting indoor localization using local feature-based deep LSTM,” *IEEE Systems Journal*, vol. 14, no. 2, pp. 3001–3010, June 2019.

[81] M. Ibrahim, M. Torki, and M. ElNainay, “CNN based indoor localization using RSS time-series,” in *Proc. 2018 IEEE Symposium on Computers and Communications*, Natal, Brazil, June 2018, pp. 1044–1049.

[82] J. Xiong and K. Jamieson, “Arraytrack: A fine-grained indoor location system,” in *Proc. ACM NSDI’13*, Lombard. IL, Apr. 2013, pp. 71–84.

[83] Y. Kang, Q. Wang, J. Wang, and R. Chen, “A high-accuracy toa-based localization method without time synchronization in a three-dimensional space,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 173–182, Jan. 2018.

[84] W. Yuan, N. Wu, Q. Guo, X. Huang, Y. Li, and L. Hanzo, “Toa-based passive localization constructed over factor graphs: A unified framework,” *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 6952–6965, Oct. 2019.

[85] A. Morar, A. Moldoveanu, I. Mocanu, F. Moldoveanu, I. E. Radoi, V. Asavei, A. Gradinaru, and A. Butean, “A comprehensive survey of indoor localization methods based on computer vision,” *MDPI Sensors*, vol. 20, no. 9, p. 2641, May 2020.

[86] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[87] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in *Proc. IEEE ICRA’11*, Shanghai, China, May 2011, pp. 3400–3407.

[88] Q. Niu, M. Li, S. He, C. Gao, S.-H. Gary Chan, and X. Luo, "Resource-efficient and automated image-based indoor localization," *ACM Transactions on Sensor Networks*, vol. 15, no. 2, pp. 1–31, May 2019.

[89] J. Luo, L. Fan, and H. Li, "Indoor positioning systems based on visible light communication: State of the art," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2871–2893, Aug. 2017.

[90] J. Armstrong, Y. A. Sekercioglu, and A. Neild, "Visible light positioning: a roadmap for international standardization," *IEEE Communications Magazine*, vol. 51, no. 12, pp. 68–73, Dec. 2013.

[91] S. Zhu and X. Zhang, "Enabling high-precision visible light localization in today's buildings," in *Proc. ACM MobiSys'17*, Niagara Falls, NY, June 2017, pp. 96–108.

[92] H. Chen, F. Li, and Y. Wang, "Echotrack: Acoustic device-free hand tracking on smart phones," in *Proc. IEEE INFOCOM'17*. Atlanta, GA, USA, 2017, pp. 1–9.

[93] Y. Liu, J. Wang, Y. Zhang, L. Cheng, W. Wang, Z. Wang, W. Xu, and Z. Li, "Vernier: Accurate and fast acoustic motion tracking using mobile devices," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 754–764, Feb. 2021.

[94] A. Damianou and N. D. Lawrence, "Deep Gaussian processes," in *Proc. 16th Int. Conf. Artificial Intelligence and Statistics*, Scottsdale, AZ, Apr./May 2013, pp. 207–215.

[95] P. Barsocchi, A. Crivello, D. La Rosa, and F. Palumbo, "A multisource and multivariate dataset for indoor localization methods based on WLAN and geo-magnetic field fingerprinting," in *Proc. IEEE IPIN'16*, Alcala de Henares, Spain, Oct. 2016, pp. 1–8.

[96] S. Pradhan *et al.*, "RIO: A pervasive RFID-based touch gesture interface," in *Proc. ACM MobiCom'17*, New York, NY, Oct. 2017, pp. 261–274.

[97] C. Yang, X. Wang, and S. Mao, "Respiration monitoring with RFID in driving environments," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 500–512, Feb. 2021.

[98] ——, "Unsupervised detection of apnea using commodity RFID tags with a recurrent variational autoencoder," *IEEE Access Journal*, vol. 7, no. 1, pp. 67 526–67 538, June 2019.

[99] ——, "RFID-Pose: Vision-aided 3D human pose estimation with RFID," *IEEE Transactions on Reliability*, vol. 70, no. 3, pp. 1218–1231, Sept. 2021.

[100] ——, "Subject-adaptive skeleton tracking with RFID," in *Proc. The 16th IEEE International Conference on Mobility, Sensing and Networking (MSN 2020)*, Tokyo, Japan, Dec. 2020.

[101] X. Wang, J. Zhang, Z. Yu, S. Mao, S. Periaswamy, and J. Patton, "On remote temperature sensing using commercial UHF RFID tags," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 715–10 727, Dec. 2019.

[102] J. Wang, J. Xiong, X. Chen, H. Jiang, R. K. Balan, and D. Fang, "Simultaneous material identification and target imaging with commodity RFID devices," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 739–753, Feb. 2021.

[103] J. Hightower, R. Want, and G. Borriello, "SpotON: An indoor 3D location sensing technology based on RF signal strength," UW CSE Technical Report, 2000.

[104] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," in *Proc.IEEE PerCom'03.*, Dallas, TX, Mar. 2003, pp. 407–415.

[105] H. Jin, Z. Yang, S. Kumar, and J. I. J. I Hong, "Towards wearable everyday body-frame tracking using passive RFIDs," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, p. Article No.: 145, Jan. 2018.

[106] C. Wang, J. Liu, Y. Chen, L. Xie, H. Liu, and S. Lu, "RF-kinect: A wearable RFID-based approach towards 3D body movement tracking," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, p. Article No: 41, Mar. 2018.

[107] 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, Jan. 2017.

[108] X. Wang, X. Wang, and S. Mao, "Indoor fingerprinting with bimodal CSI tensors: A deep residual sharing learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4498–4513, Mar. 2021.

[109] W. 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, Jan./Mar. 2020.

[110] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE CVPR'21*, June 2021, pp. 10 012–10 022.

[111] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, "Phaser: Enabling phased array signal processing on commodity WiFi access points," in *Proc. ACM Mobicom'14*, Maui, HI, Sept. 2014, pp. 153–164.

[112] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using WiFi," in *Proc. ACM SIGCOMM'15*, London, UK, Aug. 2015, pp. 269–282.

[113] X. Li, S. Li, D. Zhang, J. Xiong, Y. Wang, and H. Mei, "Dynamic-music: accurate device-free indoor localization," in *Proc. ACM UbiComp 2016*, Heidelberg, Germany, Sept. 2016, pp. 196–207.

[114] J. Oh and J. Kim, "Adaptive k-nearest neighbour algorithm for wifi fingerprint positioning," *Ict Express*, vol. 4, no. 2, pp. 91–94, June 2018.

[115] D. Li, B. Zhang, and C. Li, "A feature-scaling-based k-nearest neighbor algorithm for indoor positioning systems," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 590–597, Aug. 2016.

[116] Y. Xie, Y. Wang, A. Nallanathan, and L. Wang, “An improved k-nearest-neighbor indoor localization method based on spearman distance,” *IEEE Signal Processing Letters*, vol. 23, no. 3, pp. 351–355, Jan. 2016.

[117] L. Calderoni, M. Ferrara, A. Franco, and D. Maio, “Indoor localization in a hospital environment using random forest classifiers,” *Expert Systems with Applications*, vol. 42, no. 1, pp. 125–134, Jan. 2015.

[118] X. Guo, N. Ansari, L. Li, and H. Li, “Indoor localization by fusing a group of fingerprints based on random forests,” *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4686–4698, Dec. 2018.

[119] Y. Zhang, D. Li, and Y. Wang, “An indoor passive positioning method using csi fingerprint based on adaboost,” *IEEE Sensors Journal*, vol. 19, no. 14, pp. 5792–5800, July 2019.

[120] Z. Liu, D. Liu, J. Xiong, and X. Yuan, “A parallel adaboost method for device-free indoor localization,” *IEEE Sensors Journal*, vol. 22, no. 3, pp. 2409–2418, Feb. 2022.

[121] J.-R. Jiang, H. Subakti, and H.-S. Liang, “Fingerprint feature extraction for indoor localization,” *Sensors*, vol. 21, no. 16, p. 5434, Aug. 2021.

[122] J. Luo and L. Fu, “A smartphone indoor localization algorithm based on wlan location fingerprinting with feature extraction and clustering,” *Sensors*, vol. 17, no. 6, p. 1339, June 2017.

[123] X. Wang, X. Wang, and S. Mao, “ResLoc: Deep residual sharing learning for indoor localization with CSI,” in *Proc. IEEE PIMRC 2017*, Montreal, Canada, Oct. 2017, pp. 1–6.

[124] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[125] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *ACM ECCV’16*, Amsterdam, Nertherlands, Sept. 2016, pp. 630–645.

[126] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *ACM ECCV’16*, Amsterdam, Netherlands, Oct. 2016, pp. 483–499.

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

[128] A. Graves, N. Jaitly, and A.-r. Mohamed, “Hybrid speech recognition with deep bidirectional lstm,” in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 273–278.

[129] X. Li and X. Wu, “Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition,” in *Proc. IEEE ICASSP’15*, South Brisbane, Australia, Apr. 2015, pp. 4520–4524.

[130] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[131] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[132] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, “Tokens-to-token vit: Training vision transformers from scratch on imagenet,” in *Proc. IEEE CVPR’21*, June 2021, pp. 558–567.

[133] X. Wang, C. Yang, and S. Mao, “TensorBeat: Tensor decomposition for monitoring multi-person breathing beats with commodity WiFi,” *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 1, pp. 8:1–8:27, Sept. 2017.

[134] J. Zhang, Z. Yu, X. Wang, Y. Lyu, S. Mao, S. Periaswamy, J. Patton, and X. Wang, "RFHUI: An RFID based human-unmanned aerial vehicle interaction system in an indoor environment," *Elsevier/KeAi Digital Communications and Networks J.*, vol. 6, no. 1, pp. 14–22, Feb. 2020.

[135] G. Wang, C. Qian, K. Cui, X. Shi, H. Ding, W. Xi, J. Zhao, and J. Han, "A universal method to combat multipaths for RFID sensing," in *Proc. IEEE INFOCOM'20*, Toronto, Canada, July 2020, pp. 277–286.

[136] L. Shangguan and K. Jamieson, "The design and implementation of a mobile RFID tag sorting robot," in *Proc. ACM MobiSys'16*, Singapore, June 2016, pp. 31–42.

[137] J. Wang, D. Vasisht, and D. Katabi, "Rf-idraw: Virtual touch screen in the air using rf signals," *ACM SIGCOMM'14*, pp. 235–246, Aug. 2014.

[138] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, Jan. 2017.

[139] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE CVPR'20*, June 2020, pp. 9729–9738.

[140] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE CVPR'22*, New Orleans, LA, June 2022, pp. 16 000–16 009.

[141] Y. Tang, D. Yang, W. Li, H. R. Roth, B. Landman, D. Xu, V. Nath, and A. Hatamizadeh, "Self-supervised pre-training of swin transformers for 3d medical image analysis," in *Proc. IEEE CVPR'22*, New Orleans, LA, June 2022, pp. 20 730–20 740.

[142] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[143] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox, “3D human pose estimation in RGBD images for robotic task learning,” in *Proc. IEEE ICRA’18*, Brisbane, Australia, May 2018, pp. 1986–1992.

[144] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. Roth, and D. Xu, “Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images,” *arXiv preprint arXiv:2201.01266*, 2022.

[145] D. Vasisht, S. Kumar, and D. Katabi, “Decimeter-level localization with a single wifi access point,” in *ACM NSDI’16*, Boston, MA, Mar. 2016, pp. 165–178.

[146] A. Mittal, S. Tiku, and S. Pasricha, “Adapting convolutional neural networks for indoor localization with smart mobile devices,” in *Proc. 2018 ACM Great Lakes Symposium on VLSI*, Chicago, IL, May 2018, pp. 117–122.

[147] T. Zhang and M. Yi, “The enhancement of WiFi fingerprint positioning using convolutional neural network,” in *Proc. 2018 Int. Conf. Computer, Commun. Netw. Technol.*, Wuzhen, China, June 2018, pp. 479–483.

[148] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *arXiv preprint arXiv:1312.6199*, Dec. 2013. [Online]. Available: https://arxiv.org/abs/1312.6199

[149] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint arXiv:1412.6572*, Dec. 2014. [Online]. Available: https://arxiv.org/abs/1412.6572

[150] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proc. IEEE CVPR 2015*, Boston, MA, June 2015, pp. 1–9.

[151] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, “Adversarial attacks on medical machine learning,” *Science*, vol. 363, no. 6433, pp. 1287–1289, Mar. 2019.

[152] S. G. Finlayson, H. W. Chung, I. S. Kohane, and A. L. Beam, “Adversarial attacks against medical deep learning systems,” *arXiv preprint arXiv:1804.05296*, Apr. 2018. [Online]. Available: https://arxiv.org/abs/1804.05296

[153] S. Thys, W. Van Ranst, and T. Goedemé, “Fooling automated surveillance cameras: Adversarial patches to attack person detection,” in *Proc. IEEE CVPR’19 Workshops*, Long Beach, CA, June 2019, pp. 49–55.

[154] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, “Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition,” in *Proc. 2016 ACM SIGSAC Conf. Computer Communications Security*, Vienna, Austria, Oct. 2016, pp. 1528–1540.

[155] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” *arXiv preprint arXiv:1801.07455*, Jan. 2018. [Online]. Available: https://arxiv.org/abs/1801.07455

[156] X. Gao, W. Hu, J. Tang, P. Pan, J. Liu, and Z. Guo, “Generalized graph convolutional networks for skeleton-based action recognition,” in *Proc. 9th Int. Conf. Computing and Pattern Recognition*, Xiamen, China, Oct. 2020, pp. 43–49.

[157] J. Liu, N. Akhtar, and A. Mian, “Adversarial attack on skeleton-based human action recognition,” *arXiv preprint arXiv:1909.06500*, Sept. 2019. [Online]. Available: https://arxiv.org/abs/1909.06500

[158] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, “Deep text classification can be fooled,” *arXiv preprint arXiv:1704.08006*, Apr. 2017. [Online]. Available: https://arxiv.org/abs/1704.08006

[159] N. Papernot, P. McDaniel, A. Swami, and R. Harang, “Crafting adversarial input sequences for recurrent neural networks,” in *Proc. IEEE MILCOM 2016*, Baltimore, MD, Nov. 2016, pp. 49–54.

[160] N. Tang, S. Mao, and R. M. Nelms, “Adversarial attacks to solar power forecast,” in *Proc. IEEE GLOBECOM 2021*, Madrid, Spain, Dec. 2021, pp. 1–6.

[161] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, “Application of machine learning in wireless networks: Key technologies and open issues,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 4, pp. 3072–3108, Fourth Quarter 2019.

[162] Y. Tu, Y. Lin, C. Hou, and S. Mao, “Complex-valued networks for automatic modulation classification,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10 085–10 089, 2020.

[163] M. Sadeghi and E. G. Larsson, “Adversarial attacks on deep-learning based radio signal classification,” *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2018.

[164] Y. Lin, H. Zhao, Y. Tu, S. Mao, and Z. Dou, “Threats of adversarial attacks in DNN-based modulation recognition,” in *Proc. IEEE INFOCOM 2020*, Toronto, Canada, July 2020, pp. 2469–2478.

[165] Z. Bao, Y. Lin, S. Zhang, Z. Li, and S. Mao, “Threat of adversarial attacks on DL-based IoT device identification,” *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 9012–9024, June 2022.

[166] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[167] T. Miyato, A. M. Dai, and I. Goodfellow, “Adversarial training methods for semi-supervised text classification,” *arXiv preprint arXiv:1605.07725*, May 2016. [Online]. Available: https://arxiv.org/abs/1605.07725

[168] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” *arXiv preprint arXiv:1706.06083*, June 2017. [Online]. Available: https://arxiv.org/abs/1706.06083

[169] Y. Dong, F. Liao, T. Pang, H. Sun, J. Zhu, X. Hu, and J. Li, “Boosting adversarial attacks with momentum,” in *Proc. IEEE/CVF CVPR’18*, Salt Lake City, UT, June 2018, pp. 9185–9193.

[170] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: A simple and accurate method to fool deep neural networks,” in *Proc. IEEE CVPR 2016*, Las Vegas, NV, June-July 2016, pp. 2574–2582.

[171] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in *Proc. 2016 IEEE Symposium on Security and Privacy*, San Jose, CA, May 2016, pp. 582–597.

[172] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *Proc. 2017 IEEE Symposium on Security and Privacy*, San Jose, CA, May 2017, pp. 39–57.

[173] L. Engstrom, D. Tsipras, L. Schmidt, and A. Madry, “A rotation and a translation suffice: Fooling CNNs with simple transformations,” *arXiv preprint arXiv:1712.02779*, Dec. 2017. [Online]. Available: https://arxiv.org/abs/1712.02779

[174] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *Proc. 2017 ACM Asia Conference on Computer and Communications Security*, Abu Dhabi, United Arab Emirates, Apr. 2017, pp. 506–519.

[175] N. Papernot *et al.*, “Technical report on the cleverhans v2.1.0 adversarial examples library,” *arXiv preprint arXiv:1610.00768*, June 2018. [Online]. Available: https://arxiv.org/abs/1610.00768

[176] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial machine learning at scale,” *arXiv preprint arXiv:1611.01236*, Nov. 2016. [Online]. Available: https://arxiv.org/abs/1611.01236