Indoor Fingerprinting with Bimodal CSI Tensors: A Deep Residual Sharing Learning Approach

Xiangyu Wang, Xuyu Wang, *Member, IEEE,* and Shiwen Mao, *Fellow, IEEE*

***Abstract*—Wi-Fi based indoor fingerprinting is attracting increasing interest in the research community due to the ubiquitous access in indoor environments. In this paper, we propose ResLoc, a deep residual sharing learning based system for indoor fingerprinting using bimodal channel state information (CSI) tensor data. The proposed ResLoc system employs CSI tensor data, including the angle of arrival and amplitude, collected from a small set of training locations with known coordinates to train the proposed dual-channel deep residual sharing learning model. The proposed new model extends the traditional deep residual learning model by incorporating two or more channels and let the channels exchange their residual signals after each residual block. Unlike prior deep learning based fingerprinting schemes, ResLoc only requires for training one group of weights for all the training locations. The proposed ResLoc system is implemented with commodity Wi-Fi devices and evaluated with extensive experiments in three representative indoor environments. The experimental results validate that the proposed ResLoc system can achieve high localization accuracy using a single Wi-Fi access point in indoor environments.**

***Index Terms*—Channel state information, fingerprinting, deep learning, deep residual learning, deep residual sharing learning.**

I. Introduction

Among the various indoor localization techniques, fingerprinting-based indoor localization has attracted great interests in the community, which stores a large amount of Wi-Fi measurements from various know locations in the offline phase, and then estimates the position of a mobile device by comparing the new measurements from the device with stored measurements. Due to the low requirements on hardware and ubiquitous availability, the *received signal strength* (RSS) has been widely used in Wi-Fi based fingerprinting systems. For instance, Radar is the first RSS-based fingerprinting scheme utilizing a deterministic location estimation method [1]. Horus is also an RSS-based fingerprinting scheme, but with a probabilistic, K-nearest-neighbor (KNN) approach for location estimation [2]. Moreover, other machine learning methods have also been applied for enhanced localization performance, such as neural networks, feature-scaling-based KNN [3], support vector machine (SVM), and compressive sensing [4]. Such RSS based schemes are constrained by (i) unstable RSS values as affected by the complex indoor propagation of Wi-Fi signals, and (ii) the fact that RSS is only a coarse representation of the Wi-Fi channel.

Consequently, there is increasing interest in channel state information (CSI) based fingerprinting. Compared to RSS, CSI carries fine-grained Wi-Fi channel information, including subcarrier-level orthogonal frequency division multiplexing (OFDM) channel measurements. Moreover, CSI is much more stable over time than RSS for a given location [5]. Several IEEE 802.11n device drivers, e.g., for Intel Wi-Fi Link 5300 network interface card (NIC) [6] and the Atheros AR9580 chipset [7], are available to extract CSI from received Wi-Fi packets. Several CSI-based fingerprinting systems have exhibited better performance than RSS-based schemes. For example, FIFS exploits the weighted average of CSI amplitudes over three antennas to achieve fine-grained localization [8]. CSI amplitudes and calibrated phases information are leveraged in DeepFi [9] and PhaseFi [10], respectively. These schemes collect CSI from all the subcarriers and all the three antennas of the 802.11n NIC and learn the location features as fingerprints with a deep autoencoder model. Moreover, the BiLoc system is proposed to exploit average CSI amplitude and phase difference information by using a bimodal deep autoencoder model [11]. Although these three localization systems can achieve good localization performance with a deep learning approach, they need to build a database to store learned location features as fingerprints for *every* training location, which increases the training time and storage space requirement.

In this paper, we propose to use bimodal CSI tensor data for fingerprinting, including both estimated angles of arrival (AoA) and CSI amplitudes, that can be read from the 5GHz Wi-Fi NIC. This approach is motivated by the fact that both AoA and CSI amplitude are relatively more stable than RSS over time. Thus they can be used to effectively extract location features. Moreover, AoA and CSI amplitude are complementary to each other under different indoor environments. For example, when the line-of-sight (LOS) component is weaker than other multipath components, the CSI amplitude will be useful for improving the localization accuracy. On the other hand, if the LOS path is blocked, the amplitude will be greatly weakened, but the estimated AoA will help to strengthen the localization accuracy. In addition, we propose a new deep

Manuscript received July 12, 2020; revised September 6, 2020; accepted September 22, 2020. Date of publication XX XX, XX; date of current version XX XX, XX. This work is supported in part by the NSF under Grants ECCS-1923163 and CNS-1822055, and through the Wireless Engineering Research and Education Center (WEREC) at Auburn University. This work was presented in part at the 2017 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Montreal, Canada, Oct. 2017. (*Corresponding author: Shiwen Mao*)

Xiangyu Wang and Shiwen Mao are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA (email: xzw0042@tigermail.auburn.edu, smao@ieee.org).

Xuyu Wang is with the Department of Computer Science, California State University, Sacramento, CA 95819-6021. This work was conducted when X. Wang was pursuing a Ph.D. degree at Auburn University (email: xuyu.wang@csus.edu).

Digital Object Identifier 10.1109/JIOT.2020.3026608.

residual sharing learning model for improving the learning efficiency, which is a dual-channel deep residual learning model while the residual signals from the two channels are shared after each residual block. The deep residual learning model has been successfully applied for image recognition [12]–[14]. As in deep residual learning, the residual blocks can be stacked in the proposed dual-channel model, to increase the depth of the deep learning network, thus achieving higher learning and representation ability. The proposed method is different from the original deep residual learning model, with the novel residual sharing feature [12]–[14]. Unlike prior deep learning based fingerprinting schemes [9]–[11], the proposed method only requires for training one group of weights in the deep residual sharing network for *all* the training locations, as a classification problem in statistical learning.

In particular, we present ResLoc, a deep **Res**idual sharing learning for indoor **Loc**alization with CSI Tensors. Using the CSI data collected from received Wi-Fi packets at a training location, we can obtain CSI tensors for the training location. Moreover, a pair of tensors are created for the dual-channel deep residual sharing learning model, one for each channel using amplitude from a different antenna. In the offline training phase, all the constructed CSI tensor pairs from all training locations are used to train the proposed deep residual sharing learning model. The proposed deep learning model consists of two parallel channels, and each channel is like a regular deep residual sharing model, with an input block, a large number of stacked residual blocks, and an output block is shared by the two channels. More important, the residual signal in each channel is shared with the other channel after each residual block, to effectively exploit the CSI tensor data. We analyze the proposed deep residual sharing learning model with respect to its forward propagation and backpropagation.

In the online testing stage, The location of the mobile device is estimated using an enhanced probabilistic method. The proposed ResLoc system is implemented with commodity Wi-Fi devices and evaluated with extensive experiments in three representative indoor environments. The experimental results validate that the proposed ResLoc system can achieve high localization accuracy using a single Wi-Fi access point in typical indoor environments.

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

- We propose to use bimodal *CSI tensor* data for indoor fingerprinting. The proposed approach can effectively exploit the rich frequency and time features of the CSI data including CSI amplitude and phase difference information (which is used to estimate the AoA of Wi-Fi signals) [15].
- We propose a deep residual sharing learning model to effectively learn the location features from collected Wi-Fi bimodal CSI tensors. The proposed deep learning model inherit the advantage of deep residual learning where a large number of residual sharing blocks can be stacked as needed to increase the depth of the deep learning model, thus achieving higher learning and representation ability. It also has the unique dual-channel and residual sharing features that ensure reasonable computation complexity in training and testing. The proposed scheme is analyzed with respect to its forward propagation and backpropagation. An enhanced probabilistic method is utilized in the online testing phase for accurate location prediction.
- We implement the proposed ResLoc system with commodity 5GHz Wi-Fi NIC (i.e., Intel 5300) and conducted extensive experiments in three representative indoor environments. The experimental results demonstrate that ResLoc outperforms several baseline schemes and can achieve high location accuracy in typical indoor environments using only a single access point.

The remainder of this paper is organized as follows. Section II reviews related work. The preliminaries are illustrated in Section III. We present the ResLoc design with an analysis in Section IV and our experimental study in Section V. Then, Section VI concludes this paper.

II. Related Work

This work is closely related to the prior works on indoor fingerprinting and AoA based localization. We briefly review these two classes of related work in this section.

A. Fingerprint based Localization

Radar [1] is the first Wi-Fi RSS based fingerprinting localization system, which proves the feasibility of using Wi-Fi RSS for indoor localization. Radar collects RSS fingerprints using one or more access points and compares newly received RSS with stored ones to estimate location. To improve the precision, Horus [2] adopts a probabilistic method for location estimation. Since RSS is a coarse measurement of Wi-Fi channel, the accuracy of RSS based schemes are limited.

Compared with RSS, CSI is carries fine-grained channel information, which captures the features of each subcarrier of the Wi-Fi channel in the frequency domain. FIFS [8] is the first work utilizing CSI amplitude for indoor localization. Fingerprinting based methods require a laborious war driving to collect CSI or RSS to guarantee the high localization precision, which is hard for a spacious environment. To alleviate such burden of data collection, DeepMap [16] presents a two-layer deep Gaussian process model to construct RSS radio maps. The experimental results show the distance error does not change significantly when only 50% fingerprints are used.

Recently, with the rise of big data, deep learning techniques have shown great potential in several areas [17], [18], e.g., computer vision and natural language processing. To exploit high learning power of deep learning, several recent works use deep neural networks for indoor localization. DeepFi [19] is the first to generate fingerprints with a deep autoencoder, where CSI amplitudes are used. Similar to DeepFi [19], PhaseFi [10] uses CSI calibrated phase for indoor fingerprinting with a deep autoencoder. In addition, bimodal CSI data is utilized in BiLoc [11] with a deep autoencoder to achieve higher localization accuracy. Compared with BiLoc, in addition to the better performance as shown in our experimental study, the proposed ResLoc system only requires one common set of weights for all the training locations, i.e., it is not necessary for ResLoc to store fingerprints for every training

location like BiLoc, PhaseFi, and DeepFi do. Moreover, the proposed ResLoc system does not need a predetermined ratio for fusing the bi-modal data as in BiLoc.

Furthermore, WiDeep [20] improves the robustness of localization by combining a stacked denoising autoencoder deep learning model and a probabilistic framework. The recent works [21], [22] show that deep auotoencoder networks also contribute to device-free indoor localization. CiFi [23] is the first work to deploy Deep Convolutional Neural Network (DCNN) for indoor localization. In contrast to DeepFi that requires training weights for every training location, CiFi has the same advantage as ResLoc that it trains only one common set of weights for all the training locations. In addition to AoA, RSS and CSI amplitude are also utilized to train the DCNN model in [24]–[27], and deep learning models are also used for outdoor cellular-based localization systems [28], [29].

Deep learning methods are also exploited to reduce human effort in data collection. AF-DCGAN [30] generates CSI fingerprints with an amplitude feature, deep convolutional generative adversarial network model, which augments the fingerprint database and enhances the accuracy of localization. In addition, deep generative method such as variational autoencoder (VAE) model can be utilized for data augmentation in cellular-based localization [31].

B. AoA based Localization

To the best of our knowledge, ArrayTrack [32] is the first AoA based Wi-Fi localization system, which estimates the incoming angles using the MUSIC algorithm [33] and applies triangulation to localize the target device. However, the small number of antennas limits the accuracy of AoA based approaches. For example, an error of 20^o is reported for CUPID [34] with three antennas, which means such algorithm is not practical for mobile devices. To overcome the limitation of small number of antennas, SpotFi [35] applies spatial smoothing on the CSI matrix to increase the angle resolution and utilizes a enhanced MUSIC algorithm to estimate AoA and ToF simultaneously. Moreover, WiDeo [36] and ROArray [37] retrieve ToA and AoA with sparse recovery. IndoTrack [38] estimates the absolute trajectory of the mobile device by combining doppler velocity and AoA spectrum. In Ubicarse [39], a mobile device is used to emulate a large antenna array. Thus it allows to perform Synthetic Aperture Radar (SAR) with a handheld device and applies to device-free localization.

AoA estimation has also been leveraged in several other device-free localization works. MaTrack [40] proposes a novel Dynamic-MUSIC method, which is able to detect reflected signals from a moving human body and estimates the AoA of the reflected signal for localization. Typically AoA estimation relies on CSI phase. However, the authors in [41] present a magnitude-based AoA estimation algorithm, which achieves a mean absolute error of about 2.5^o. The algorithm enables AoA based localization for mobile devices that cannot gather CSI phase information. Even though AoA based algorithms achieve high precision, the computational cost of such algorithms limits their popularity on mobile devices and for real-time applications. To get rid of the high computational cost of the MUSIC algorithm, authors in [42] show that AoA and ToF can be estimated independently using the one-dimensional Modified Matrix Pencil (MMP) algorithm, which makes it possible for AoA estimation on resource-constrained mobile devices.

III. CSI and CSI Tensors

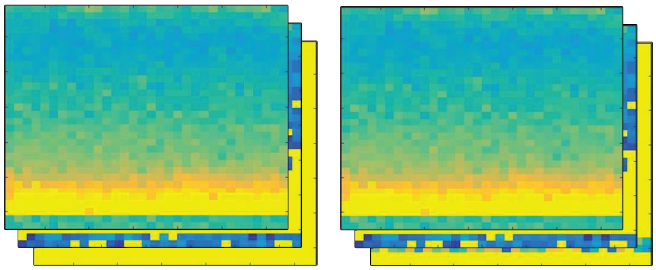
In many wireless network standards (such as LTE, WiMAX, and Wi-Fi), OFDM is used to combat frequency selective channels and achieve high spectrum efficiency. With OFDM, the spectrum channel is divided into multiple orthogonal subcarriers to alleviate channel fading and large delay spreads. Wireless data is transmitted on all the subcarriers using Inverse Fast Fourier Transform (IFFT) at the transmitter and recovered using Fast Fourier Transform (FFT) at the receiver. Cyclic redundancy is employed at the receiver to reduce the complexity caused by FFT processing.

Several 802.11n open-source device drivers provide an interface to read CSI data from several off-the-shelf Wi-Fi NICs, including the Intel Wi-Fi Link 5300 NIC [6] and the Atheros AR9580 chipsets [7]. ResLoc is implemented with the Intel 5300 NIC (with three antennas) to read 5GHz Wi-Fi CSI data from 30 out of the 56 subcarriers for a 20MHz or 40MHz channel. Let H_i denote the CSI value on subcarrier i, as a complex value given by

$$H_i = |H_i|e^{j\angle H_i}, \tag{1}$$

where $|H_i|$ and $\angle H_i$ are the amplitude and phase response of subcarrier i, respectively. ResLoc uses both CSI amplitude and phase, and translates the latter to phase difference and then to AoA, for training and testing. CSI phase difference has been shown to be quite stable over time for a given location, and contain rich location features [11], [43], [44].

To construct CSI tensors for ResLoc to use, we collect and compute bimodal CSI data including estimated AoAs and CSI amplitudes. In addition to the amplitude data read from the three antennas, we also estimate the corresponding AoA values between each pair of adjacent antennas for each subcarrier and each received packet, by using the method proposed in the BiLoc system [11]. Then, for every 30 received packets and the corresponding CSI data collected from 30 subcarriers, we can construct a CSI tensor including three images, each of which has the same size of 30×30, whose elements are the estimated AoA values and CSI amplitude values. Specifically, two of the images contain the estimated AoA values between antennas 1 and 2, and antennas 2 and 3, respectively. The third image is formed with amplitude values from one of the three antennas. Thus, from 990 received Wi-Fi packets at a given location, we can construct 33 CSI tensors for the location. In the ResLoc system, we construct CSI tensor pairs for the dual-channel deep residual sharing learning model. The difference between the two CSI tensors in a pair is that they have different amplitude images from two different antennas. For example, the amplitudes from antenna 1 is used to construct the third image in one tensor, and the amplitudes from antenna 2 is used to construct the third image in the other tensor in the dual-channel tensor pair. Fig. 1 shows the CSI tensors for the



(a) Channel 1 (b) Channel 2

Fig. 1. The constructed CSI tensors using AoA and CSI amplitude measurements.

two channels, which are fed into the two channels in the dual-channel deep residual sharing learning model, respectively. We can see that for these two CSI tensors, their first two images are identical while their third images are different.

The use of CSI tensors in ResLoc is motivated by three observations. First, by using the three dimensional data, one can greatly strengthen the performance of the deep learning model for the classification problem in indoor fingerprinting. Second, the data read from all the 30 subcarriers from all the packet samples can be utilized in the three images of the CSI tensor, which carry rich frequency and time features of the Wi-Fi propagation channel. We can thus learn more useful features from the CSI tensor. Third, the two types of CSI data, i.e., estimated AoA and CSI amplitude, are complementary to each other under different indoor environments. Thus leveraging the bimodal CSI data can lead to robust localization performance.

IV. The Proposed ResLoc System Design

A. The ResLoc System Architecture

The ResLoc system architecture is presented in Fig. 2, which consists of one transmitter, which is a mobile Wi-Fi device, and one receiver, which is a Wi-Fi access point. Both Wi-Fi devices are equipped with the Intel 5300 NIC. To collect CSI data, the transmitter is set to the *injection* mode, and the receiver is set to the *monitor* mode. The receiver's NIC reports CSI measurements from 30 out of the 56 subcarriers from each of the three antennas. After collecting the CSI data, we next create dual-channel CSI tensor pairs using estimated AoAs and CSI amplitudes. ResLoc is an indoor fingerprinting method, which includes an offline training stage and an online location estimation stage. As discussed, we propose a new deep residual sharing learning model for ResLoc, which is trained with the dual-channel CSI tensor pairs to find the optimal weights of the deep learning network. For online location estimation, we feed newly collected CSI tensor pairs from the device, whose location is to be determined, into the well-trained deep learning model to estimate the location of the mobile device, using an enhanced probabilistic approach.

The proposed ResLoc system is quite different from traditional fingerprinting methods, which build a database for every training location with either raw data or learned features as fingerprints [1], [9], [10], [19], [45]. In fact, the ResLoc system only needs to train one group of weights for the deep residual sharing model for all the training locations like a regression

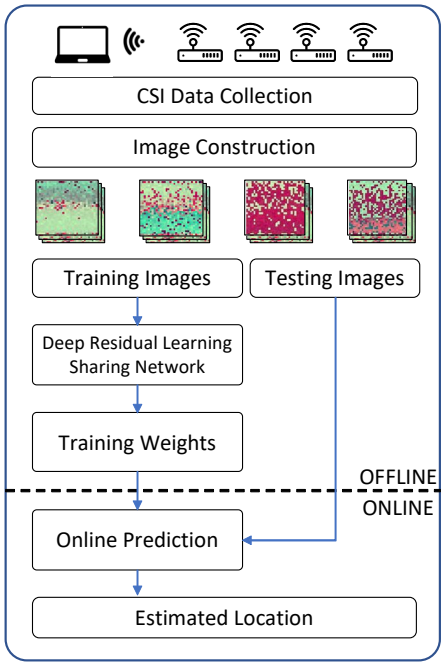


Fig. 2. The proposed ResLoc system architecture.

or classification problem in statistical learning. Furthermore, the proposed method contributes to improving the robustness of indoor localization, by effectively learning and representing the rich location features hidden in bimodal Wi-Fi CSI data.

B. Deep Residual Sharing Learning Design

The proposed deep residual sharing learning model consists of the input block, the residual blocks, and the output block, as shown in Fig. 3. We examine the design of these building blocks and analyze the forward propagation and backpropagation in the following.

1) Input Block: The input block consists of four different layers, which are: the Convolution2D layer, the Batch Normalization layer, the Activation layer, and the Max Pooling layer. The input block can learn the local dependency and scale-invariant features from bi-modal CSI tensors. Furthermore, the input block can create a more abstract representation of the input CSI tensor data across the layers, to enhance feature extraction from CSI tensor data for indoor fingerprinting. We discuss the four building-block layers in the following.

Unlike Convolution1D that is mainly used for 1D data, ResLoc uses the Convolution2D layer to obtain feature maps within local regions in the input CSI tensor, or the previous layer's feature maps, with several convolution kernels. The Convolution2D layer can exploit the rich frequency and time features in bimodal CSI tensor data. The convolution operation with weights sharing can improve the efficiency of training the deep learning model. It can also improve the data representation ability to reduce the localization error. In several prior works, the Convolution2D layer has been shown effective for indoor localization based on Wi-Fi RSS [25]–[27] and CSI images [23], [24], which motivate us to leverage the Convolution2D layer for the proposed ResLoc system. In addition, the Batch Normalization layer can adjust the input distribution

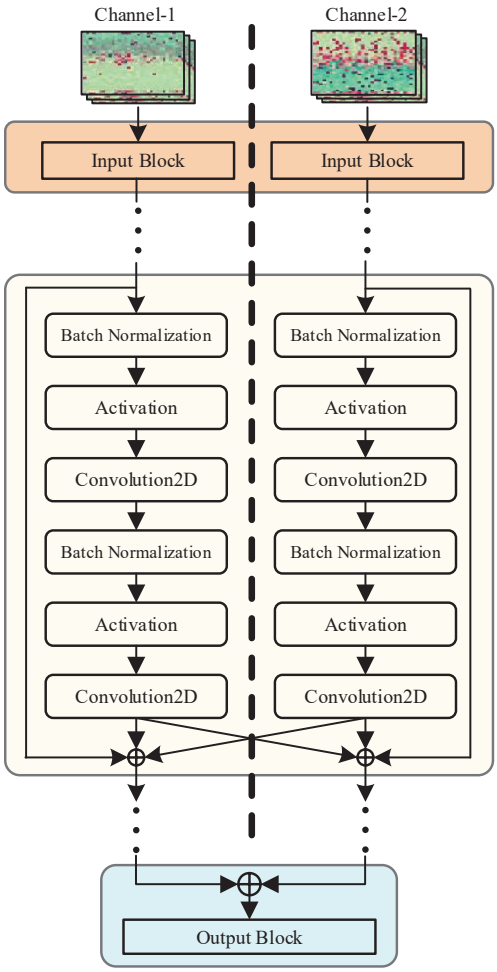


Fig. 3. The proposed deep residual sharing learning model.

and thus alleviate the problem of Internal Covariance Shift, which can mitigate overfitting in training [46]. The Activation layer incorporates the *rectified linear unit* (ReLU) activation function, to achieve faster convergence than traditional *sigmoid* and *tanh* functions [47]. The max pooling layer is to reduce the resolution of feature maps by downsampling over a local neighborhood in the feature maps of the previous layer.

2) Residual Block: The core of deep residual sharing is multiple stacked residual blocks. We propose a new deep residual sharing learning model for improving the training performance with bimodal, dual-channel CSI tensor pairs. The proposed method is different from the original deep residual learning model, by introducing the dual-channel structure and the residual sharing feature. Moreover, we can stack as many residual blocks as needed to increase the depth of the deep network, thus achieving stronger learning and representation ability.

The main idea of traditional deep residual learning is that, instead of learning the underlying mapping $F(x)$ (with input x) using stacked layers, it learns the *residual function* defined as [13], [14]

$$R(x) = F(x) - x. \tag{2}$$

The original mapping can be obtained by $F(x) = R(x) + x$ at

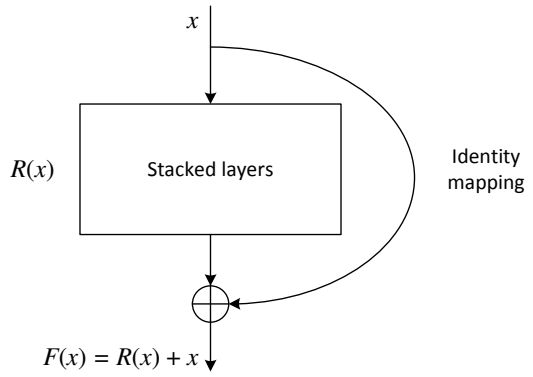


Fig. 4. Illustration of the residual learning concept [13].

the output of the stacked layers, while x is available through an *identity mapping* with a shortcut connection, as shown in Fig. 4. Such a structure allows stacking a large number of layers, which is also easier to train than simply stacking a large number of layers. With deep residual learning, one or more stacked layers can be bypassed by setting the weights in the layers to zero.

Motivated by the huge success of deep residual learning in image recognition [13], [14], we propose a dual-channel model as shown in Fig. 3. The proposed deep residual sharing learning model consists of two identical channels. Input tensor pairs are fed into the channels, one tensor for each channel. Note that since the two channels are trained with different data, they may have different weights. At the output of a residual block, the residual signals are shared between the two channels. The residual function includes two layers of convolution operations, each of which consists of the batch normalization layer, the activation layer, and the convolution2D layer. These layers are implemented similarly as that in the input block.

Next we provide an analysis of deep residual sharing learning with respect to its forward propagation and back-propagation. Let x_k^1 and x_k^2 denote the input data to the kth residual block of channel 1 and channel 2, respectively, each with two 3×3 convolution layers. Let $R(\cdot)$ denote the residual function. According to Fig. 3, we have

$$x_{k+1}^1 = x_k^1 + R(x_k^1) + R(x_k^2) \tag{3}$$
$$x_{k+1}^2 = x_k^2 + R(x_k^2) + R(x_k^1), \tag{4}$$

for the $(k+1)$th residual block in channel 1 and channel 2, respectively. Assume the signals propagate from layer k to layer K $(K > k)$. We can recursively obtain x_K^1 and x_K^2 for the Kth residual block in channel 1 and channel 2, respectively, which are given by

$$x_K^1 = x_k^1 + \sum_{i=k}^{K-1} R(x_i^1) + \sum_{i=k}^{K-1} R(x_i^2) \tag{5}$$
$$x_K^2 = x_k^2 + \sum_{i=k}^{K-1} R(x_i^2) + \sum_{i=k}^{K-1} R(x_i^1). \tag{6}$$

From the above forward propagation equations (5) and (6), we can see that the outputs at layer K, i.e., x_K^1 and x_K^2, share the same accumulated residual function, which is represented by the summation of all the preceding residual functions plus

inputs x_i^1 or x_i^2, respectively. This particular form helps to reduce the error of gradient propagation. Moreover, it is easier to train the residual sharing functions, each of which can be forced to zero when the identity mapping is optimal, thus greatly reducing the training time.

We next consider backpropagation. Let $\mathcal{L}$ denote the loss function. Based on the chain rule of backpropagation, from layer K to layer k, we have that

$$\frac{\partial \mathcal{L}}{\partial x_k^1} = \frac{\partial \mathcal{L}}{\partial x_K^1}\left(1 + \frac{\partial}{\partial x_k^1}\left(\sum_{i=k}^{K-1}\left(R(x_i^1) + R(x_i^2)\right)\right)\right) \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial x_k^2} = \frac{\partial \mathcal{L}}{\partial x_K^2}\left(1 + \frac{\partial}{\partial x_k^2}\left(\sum_{i=k}^{K-1}\left(R(x_i^2) + R(x_i^1)\right)\right)\right). \tag{8}$$

From the above backpropagation equations, it can be seen that the gradients $\frac{\partial \mathcal{L}}{\partial x_K^1}$ and $\frac{\partial \mathcal{L}}{\partial x_K^2}$ are directly propagated back to any shallower inputs x_k^1 and x_k^2. Moreover, because the gradients for the residual sharing functions, i.e.,

$$\frac{\partial}{\partial x_k^1}\left(\sum_{i=k}^{K-1}\left(R(x_i^1) + R(x_i^2)\right)\right) \tag{9}$$

$$\frac{\partial}{\partial x_k^2}\left(\sum_{i=k}^{K-1}\left(R(x_i^2) + R(x_i^1)\right)\right), \tag{10}$$

are not always "-1," it would be hard to cancel the *additive* gradients $\frac{\partial \mathcal{L}}{\partial x_K^1}$ and $\frac{\partial \mathcal{L}}{\partial x_K^2}$ for the mini-batch by stochastic gradient descent (SGD). Therefore, the problem of *vanishing gradient* can be effectively mitigated. Therefore, the proposed deep residual sharing learning model can effectively increase the learning ability and better exploit the bimodal, dual-channel CSI tensor data.

3) Output Block: In the output block, the output signals from the two channels are added to obtain a single signal. Several basic data operations are then performed, including batch normalization, ReLU activation, and max pooling. The main operation in the output block is a fully-connected layer, which employs a basic neural network with one hidden layer, followed by a *softmax* classifier. The input to the softmax function is an N-dimensional vector $\vec{z} = [z_1, z_2, ..., z_N]$, where N is the number of clusters (i.e., the number of known training locations). The softmax function transforms the N-dimensional vector $\vec{z}$ into a normalized vector $\vec{p} = [p_1, p_2, ..., p_N]$, whose elements are given by

$$p_i = \frac{e^{z_i}}{\sum_{n=1}^{N} e^{z_n}}, \quad \text{for } i = 1, 2, ..., N. \tag{11}$$

For ResLoc, we define the loss function $\mathcal{L}$ as the cross-entropy to measure the difference between the normalized prediction results and the true labels, that is

$$\mathcal{L} = -\sum_{n=1}^{N} y(n) \cdot \log(p_n), \tag{12}$$

where $y(n)$ represents the true label of the training data for the nth training location. The parameters in the deep network are trained with the SGD method by minimizing the loss function. The weight training procedure will be discussed in detail in Section IV-C.

4) Model Structure: Fig. 5 shows the detail model structures for different ResLoc models. In ResLoc, the input block provides preliminary processing, such as Convolution2D, Batch Normalization, Activation, and Max Pooling, for raw CSI tensors. For example, the down-sampling in the input block is performed with a stride of 2 in Convolution2D. The body of the network is constructed with 4 types of residual blocks. The detailed structure of the residual block is as shown in Fig. 3. The Batch normalization layers and Activation layers are not shown in this figure due to limited space.

Each type of the residual blocks has a different number of convolution kernels, and it repeats for different times for different ResLoc configurations. In ResLoc, the size of all the convolutional kernels is set as 3×3. The number of convolutional kernels is different in different type of basic blocks. For example, we consider the following form $\left[\begin{smallmatrix}3\times3,\ 64\\ 3\times3,\ 64\end{smallmatrix}\right] \times 3 \times 2$, which means that this type of block includes two convolutional layers with size 3×3, and each layer is comprised of 64 convolutional kernels. It also indicates that this block repeats for three times in both channels. For the configuration of 2-2-2-2, each type of residual blocks repeats twice to comprise the body of the ResLoc system. A merge layer is inserted between each basic block. Except for the merge layer between different types of basic blocks, such as the merge layer between Block 1 and Block 2 in the figure, a merge layer also exists between the same type of basic blocks, which is not shown in the figure. Between residual block 4 and the output block, a convolution2D layer, composed of 1024 kernels, is inserted.

Finally, the output result is obtained with the output block, which includes a batch normalization layer, a ReLu activation layer, a max pooling layer, a fully-connected layer, and a softmax layer. With the help of Keras, we are able to count the number of parameters for each configuration with the built-in function $model.count\_params()$. The numbers of parameters are listed in the last row of the figure.

C. Offline Weight Training Procedure

The pseudocodes for offline training with a pair of input CSI tensors are presented in Algorithm 1 and Algorithm 2. The inputs to Algorithm 1 are two bimodal CSI tensors. Each input tensor consists of two AoA slices and an amplitude slice, with size 30×30. The elements are either the estimated AoA or the CSI amplitude extracted from every 30 received Wi-Fi packets (in the x-axis dimension) and from the 30 subcarriers (in the y-axis dimension).

In Algorithm 1, the input datasets are split into mini batches to train the deep learning model. First, batches are processed by the input block. To obtain the output of the input block, batches are handled by the input block layers sequentially (Lines 8–11 in Algorithm 1). Because of the dual-channel architecture, each of the two input tensors passes through one of the two channels in parallel (as implemented in TensorFlow). The outputs of the input block are then processed by the residual blocks. Then the outputs of residual blocks are fed into the output block. In addition to the similar modules as in the input block, the output block also includes two special layers, a merge layer and a fully-connected layer. Before the output

Type of Blocks	1-1-1-1	2-2-2-2	3-3-3-3	4-4-4-4	5-5-5-5	6-6-6-6
Input Block	7×7, 64, stride 2					
	Max pooling					
Residual Block 1	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix} \times 1 \times 2$	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix} \times 2 \times 2$	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix} \times 3 \times 2$	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix} \times 4 \times 2$	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix} \times 5 \times 2$	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix} \times 6 \times 2$
Residual Block 2	Merge layer					
	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix} \times 1 \times 2$	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix} \times 2 \times 2$	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix} \times 3 \times 2$	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix} \times 4 \times 2$	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix} \times 5 \times 2$	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix} \times 6 \times 2$
Residual Block 3	Merge layer					
	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix} \times 1 \times 2$	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix} \times 2 \times 2$	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix} \times 3 \times 2$	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix} \times 4 \times 2$	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix} \times 5 \times 2$	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix} \times 6 \times 2$
Residual Block 4	Merge layer					
	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix} \times 1 \times 2$	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix} \times 2 \times 2$	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix} \times 3 \times 2$	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix} \times 4 \times 2$	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix} \times 5 \times 2$	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix} \times 6 \times 2$
	Merge layer					
	$[1 \times 1, 1024] \times 1 \times 2$	$[1 \times 1, 1024] \times 1 \times 2$	$[1 \times 1\ 1024] \times 1 \times 2$	$[1 \times 1, 1024] \times 1 \times 2$	$[1 \times 1, 1024] \times 1 \times 2$	$[1 \times 1, 1024] \times 1 \times 2$
Output Block	Merge layer					
	Max pooling, 1024-d fc,softmax					
Number of Parameters	10896916	23449876	36002836	48555796	61108756	73661716

Fig. 5. The proposed ResLoc model structure.

signals from the residual blocks pass the merge layer, they are processed again by batch normalization, activation, and convolution in parallel (Lines 15–18). The outputs from the two channels are next merged in the merge layer. The output of the merge layer is then processed by batch normalization, activation, pooling, and a fully connected layer sequentially (Lines 21–24). Once the output of the fully-connected layer is obtained, the cross entropy between the prediction results and the labels is computed. Then, the weights and biases are updated using the error with backpropagation. Finally, we need to update all the batches, which is implemented for 50 epochs in the offline training algorithm.

The pseudocode for the residual blocks is presented in Algorithm 2. The inputs to the algorithm are the number of repetitions of residual blocks K and the two output signals from the input blocks, i.e., I_1 and I_2. The repetition number K defines the number of residual blocks that are stacked to form a deep residual block. Typically, the repetition is a one-dimensional array, whose length is the number of residual blocks with different numbers of convolution layers. And the elements of the array define the sizes of residual blocks with the same-size convolution layers. The total amount of residual blocks is defined by the sum of elements in the repetition vector. The basic residual block is composed of a dual-channel structure, which includes two convolution layers, a batch normalization layer, and an activation layer in each channel, as shown in Fig. 3. The stacking order of these layers is as that given in Lines 7–10. It is worth noting that there is a sharing layer at the end of each residual block. The output of the previous layer and the residual output of the blocks are sum up in the sharing layer as the new output and the new residual output of the residual block (as given in Lines 12–13).

D. Online Test Procedure

In the online test stage, newly received CSI tensors from the unknown test location are fed into the well trained deep network, and ResLoc utilizes a probabilistic method to estimate the location of the mobile device. Let T denote the number of new CSI tensors collected from the unknown location, and p_{ij} denote the output result from the deep learning model corresponding to the jth CSI tensor for the ith location. Let matrix $\boldsymbol{P}$ denote the prediction output of the deep network with T new CSI tensors for the N training locations. We have

$$\boldsymbol{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1T} \\ p_{21} & p_{22} & \cdots & p_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NT} \end{bmatrix}. \tag{13}$$

To reduce the variance of the output results, the T output values for every training location are averaged out. Thus, we can obtain a vector $\vec{\bar{p}} = [\bar{p}_1, \bar{p}_2, ..., \bar{p}_N]$, where $\bar{p}_i$ is the mean of the output row vector $[p_{i1}, p_{i2}, ..., p_{iT}]$.

Finally, the location of the mobile device is estimated as a weighted average of all the N training locations, that is

$$\hat{l} = \sum_{i=1}^{N} l_i \cdot \bar{p}_i, \tag{14}$$

where l_i is the ith training location.

V. Experimental Study

A. Experiment Configuration

To evaluate the performance of ResLoc, we implement it with commodity 5GHz Wi-Fi devices. A desktop computer and a Dell laptop are configured as access point and mobile device, respectively. Both computers are equipped with an Intel 5300 NIC, with the Ubuntu desktop 14.04 LTS system.

Algorithm 1: Weight Training Algorithm

Input: Input tensor datasets T_1 and T_2, the number of repetitions of residual blocks K;
Output: Trained weights W and b;

Divide input datasets T_1 and T_2 into a batches that each contains q CSI tensors;
Let c be the channel index and M be the CSI tensor batch;
while $epoch < 50$ **do**
  for $d = 1 : a$ **do**
    $\theta_1 = M_1^a$;
    $\theta_2 = M_2^a$;
    for $c = 1 : 2$ **do**
      $\theta_c = Convolution(\theta_c)$;
      $\theta_c = BatchNormalization(\theta_c)$;
      $\theta_c = ReLU(\theta_c)$;
      $\theta_c = pool(\theta_c)$;
    end
    Execute Residual Block Algorithm 2;
    for $c = 1 : 2$ **do**
      $\theta_c = X_c$;
      $\theta_c = BatchNormalization(\theta_c)$;
      $\theta_c = ReLU(\theta_c)$;
      $\theta_c = Convolution(\theta_c)$;
    end
    $S = \theta_1 + \theta_2$;
    $S = BatchNormalization(\theta_c)$;
    $S = ReLU(S)$;
    $S = pool(S)$;
    $q = softmax(W * flattened(S) + b)$;
    Loss function $\mathcal{L} = -\sum_n y_n \cdot \log(q_n)$;
    Update weights and bias using the error with backpropagation;
  end
end

Algorithm 2: Residual Block Algorithm

Input: Output from the input block, I_1 and I_2, and number of repetitions of residual blocks K;
Output: Output of residual blocks, X_1 and X_2;

for $k = 1 : K$ **do**
  if $k == 1$ **then**
    $X_1 = I_1$;
    $X_2 = I_2$;
  end
  for $c = 1 : 2$ **do**
    $\theta_c = Convolution(X_c)$;
    $\theta_c = BatchNormalization(\theta_c)$;
    $\theta_c = ReLU(\theta_c)$;
    $\theta_c = Convolution(\theta_c)$;
  end
  $X_1 = \theta_1 + \theta_2 + X_1$;
  $X_2 = \theta_1 + \theta_2 + X_2$;
end

To transmit random Wi-Fi packets, the Dell laptop uses one antenna and works in the injection mode. The desktop is configured to operate in the monitor mode and uses three antennas to receive packets and extract CSI data. The distance between two adjacent antennas on the Desktop NIC is set to 2.68cm, which is a half wave length for the 5.58GHz Wi-Fi channel. Moreover, the physical layer is the IEEE 802.11n OFDM system with QPSK modulation and 1/2 coding rate. To accelerate the training process, we implement the offline stage of the ResLoc in Keras with a TensorFlow backend on a PC with Intel(R) Core(TM) i7-6700K CPU and a Nvidia GTX1070 GPU [48].

ResLoc is compared with three existing deep learning based fingerprinting schemes in our experiments. The first two are BiLoc [11] and DeepFi [9]. Both these baseline schemes use an autoencoder to process CSI data. We also implement a standard deep residual learning based version of ResLoc (i.e., a single channel model) and use it as a third baseline scheme. For the sake of fairness, the same CSI training dataset and testing dataset are used by all the schemes. We examine them in three typical experimental environments, including a computer laboratory, a single corridor, and two corridors.

1) *Computer Laboratory:* We set up the first testbed in a 6×9 m^2 computer laboratory in Broun Hall in the Auburn University campus. This laboratory is a cluttered environment. The furniture and appliances block most of the LOS paths. The floor plan is shown in Fig. 6, where the 15 training locations are marked as red squares, and the 15 testing locations are marked as green dots. The distance between two adjacent training locations is 1.8m. The receiver (i.e., the AP) is fixed on the table. We collect CSI data from 1000 received Wi-Fi packets for every training location in the training stage, and from 1000 received packets for every testing location in the testing stage. The number of layers for the proposed deep network is set to 34, which achieves a higher localization accuracy and smaller training time as shown in our experiments.
2) *Single Corridor:* The second test scenario is a long corridor in Broun Hall, which is 9×25 m^2 (including the rooms on both sides of the corridor). The corridor is empty with no obstacles. In this scenario, the LOS component is dominant. We choose 15 training locations and 15 testing locations arranged along a straight line. The distance between two adjacent training locations is 1m. The floor plan is shown in Fig. 7, where the red squares are training locations and the green dots are testing locations. The receiver is put at the center of the corridor. Again, CSI data is collected from 1000 packets for every training location and every testing location. The number of layers in the deep learning model for the corridor scenario is the same as that in the computer laboratory scenario.
3) *Two Corridors:* The third scenario is a 2.4×24 m^2 corridor and a 9×22 m^2 corridor, as shown in Fig. 8. In this scenario, the two corridors are empty with no obstacles around. In addition, there are strong LOS paths. 20

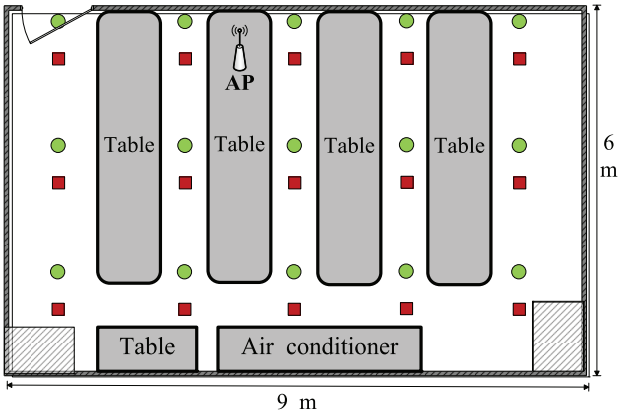


Fig. 6. Layout of the computer laboratory: training locations are marked as red squares; testing locations are marked as green dots.

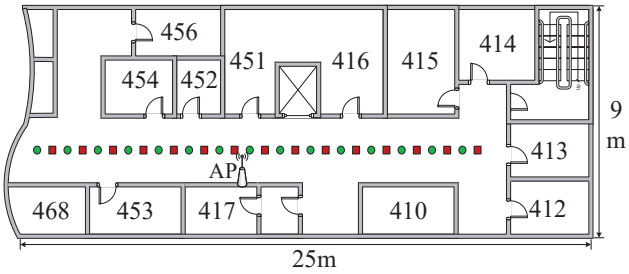


Fig. 7. Layout of the corridor: training locations are marked as red squares; testing locations are marked as green dots.

training positions are chosen (the red squares arranged along the two corridors). The distance between two adjacent training locations is 1.8m. The test locations are randomly chosen in the two corridors (different from the training positions), which are not shown in Fig. 8. As in the previous scenarios, we measure 1000 packets to obtain CSI data for each training position and collect 1000 packets for each test position. The parameters in the proposed deep learning model are the same as that in the above two scenarios.

B. Accuracy of Location Estimation

Fig. 9 presents the loss function value over epochs when training ResLoc for the laboratory and single corridor scenarios. To prevent overfitting and to reduce training time, the epoch is set to 50. As shown in Fig. 9, the training loss for the corridor scenario reaches 0.3 and the training loss for the lab scenario converges to 0.5 after 50 epochs. With the Nvidia GTX1070 GPU, the training times for the laboratory and corridor scenarios are sufficiently low, which are 608.14s and 619.35s, respectively. The test time for the laboratory and corridor scenarios are 0.587s and 0.647s, respectively, which are adequate for realtime indoor localization and navigation.

Fig. 10 presents the cumulative distribution function (CDF) of distance errors for all the 15 testing positions in the laboratory experiment. Unlike the corridor scenario that the LOS is dominant, the furniture and appliances in the lab block most of the LOS paths. The figure shows that the

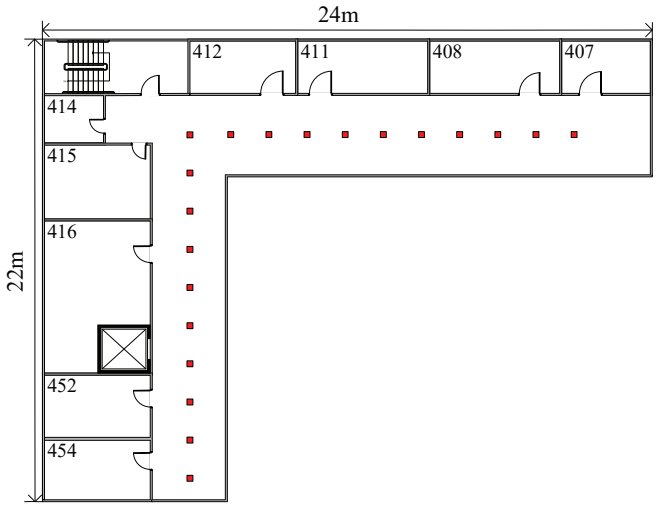


Fig. 8. Layout of the two corridors: training locations are marked as red squares.

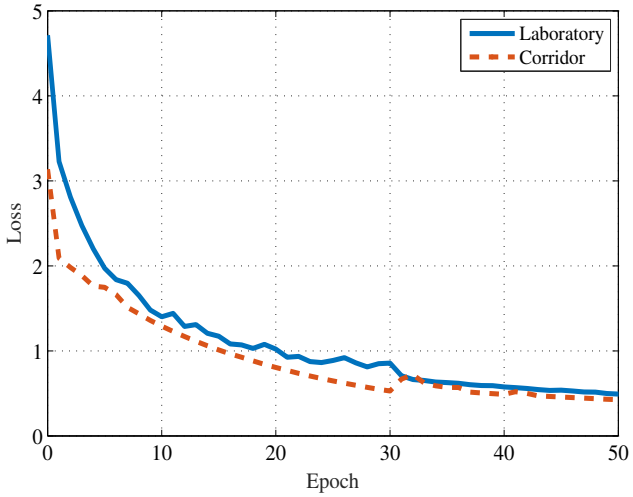


Fig. 9. Training errors for the laboratory and corridor experiments.

maximum distance errors for ResLoc with two channels and single channel are 2.46m and 2.55m, respectively, which are both lower than that of DeepFi (2.86m) and BiLoc (3.02m). In addition, the median of distance errors for ResLoc with two channels and single channel are both 0.89m, which also outperforms BiLoc and DeepFi by a reduction of 0.51m and 0.89m, respectively. For the two-channel ResLoc, 30% of the distance errors are less than 0.3m, while there is no error falling within this range for DeepFi and BiLoc. In summary, based on the proposed deep residual sharing learning model, the dual-channel ResLoc achieves the best performance in this multipath-rich scenario.

Fig. 11 plots the CDF of localization errors for the single corridor scenario. As shown in Fig. 11, the maximum distance error for ResLoc with two channels and single channel are 3.14m and 3.95m, respectively, which are significantly less than that of the other two baseline schemes. This verifies that ResLoc is more stable than DeepFi and BiLoc. In addition, the median errors for ResLoc with two channels and single

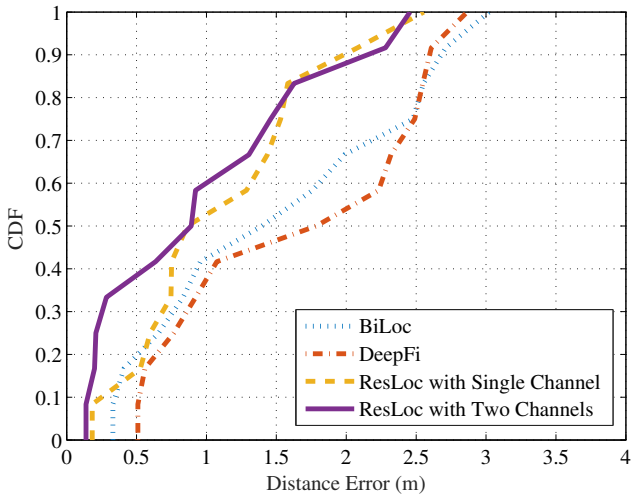


Fig. 10. CDF of localization errors for the laboratory experiment.

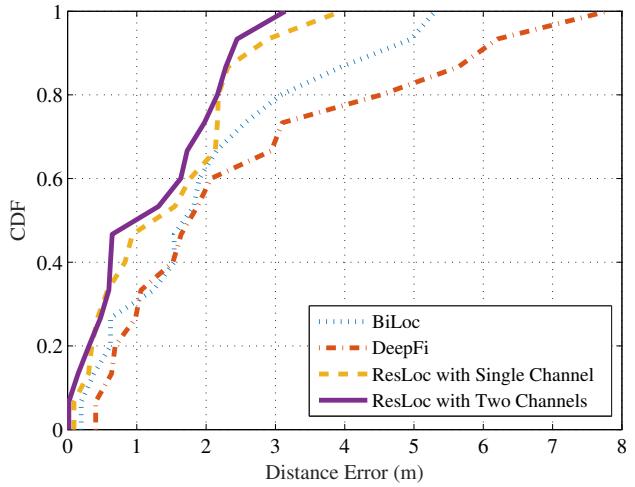


Fig. 11. CDF of localization errors for the corridor experiment.

channel, BiLoc, and DeepFi are 0.98m, 1.24m, 1.68m, and 1.75m, respectively. ResLoc with two channels achieves the best performance again in this scenario. In addition to the better performance, the ResLoc system only requires one set of weights for all training locations, i.e., ResLoc does not need to store fingerprints for every training location as BiLoc and DeepFi do. ResLoc also does not need a predetermined ratio for fusing the bi-modal data as in BiLoc.

Fig. 12 plots the CDF of localization errors in the three scenarios achieved by the proposed ResLoc method. It can be seen that ResLoc achieves the best performance in the laboratory scenario. The maximum distance errors of ResLoc in the laboratory, single-corridor, and two-corridor cases are 2.46m, 3.14m, and 3.25m, respectively. The median distance errors of ResLoc in the laboratory, single-corridor, and two-corridor cases are 0.89m, 0.98m, and 1.2m, respectively. Although ResLoc has a larger localization error in the two-corridor scenario because of the larger area, its localization accuracy is still sufficient with a median error of 1.2m. Also note that these results are achieved by using only a single AP, unlike previous RSSI-based schemes

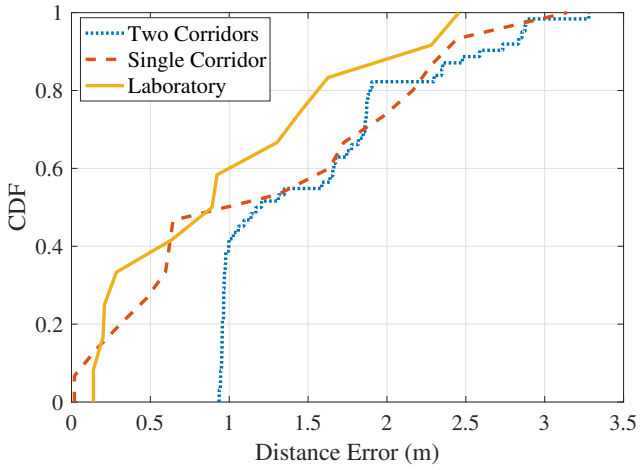


Fig. 12. CDF of localization errors for the three scenarios.

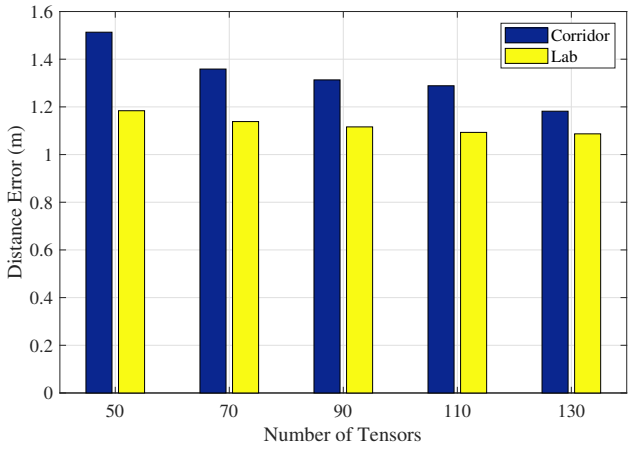


Fig. 13. The mean distance error for different numbers of CSI tensors.

C. Effect of Design Parameters

To evaluate the impact of different parameters of the proposed deep residual sharing learning model on indoor localization, we focus on the two scenarios, i.e., the lab and single corridor, in this section.

1) Impact of the Number of Tensors: To explore how the number of tensors affects the distance error, we create five datasets with different numbers of tensors for every location. As shown in Fig. 13, the distance error declines with the increase of the number of tensors. The lowest distance errors, 1.09m for the lab case and 1.18m for the corridor case, are achieved when the number of tensors is 130. This result indicates that the number of tensors is positively related to the localization accuracy. Furthermore, the distance error in the corridor is more sensitive to the number of tensors. We also notice that all the distance errors in the lab case are smaller than 1.2m and the distance errors in the corridor are lower than 1.3m when the number of tensors is greater than 90.

Fig. 14 presents the training times of all the datasets with different numbers of tensors. It is not surprising to see that the training time is directly proportional to the number of tensors. For the same number of tensors, the training time of the corridor case is slightly longer than that of the lab case.

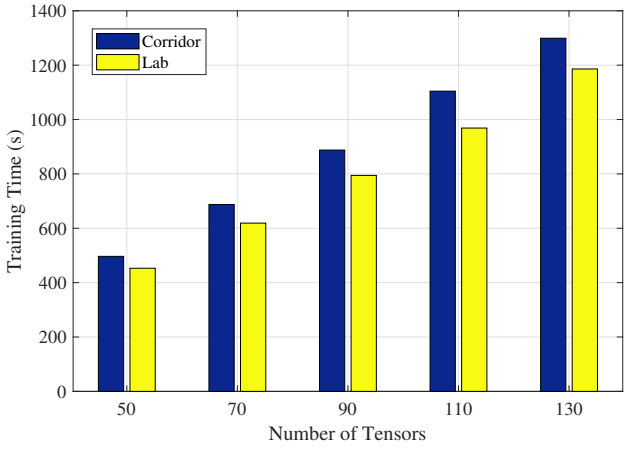


Fig. 14. The average training time for different number of tensors.

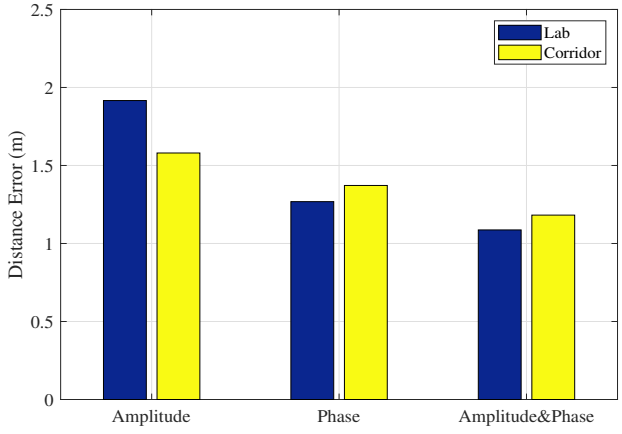


Fig. 15. The average distance errors for different datasets with the dual-channel ResLoc system.

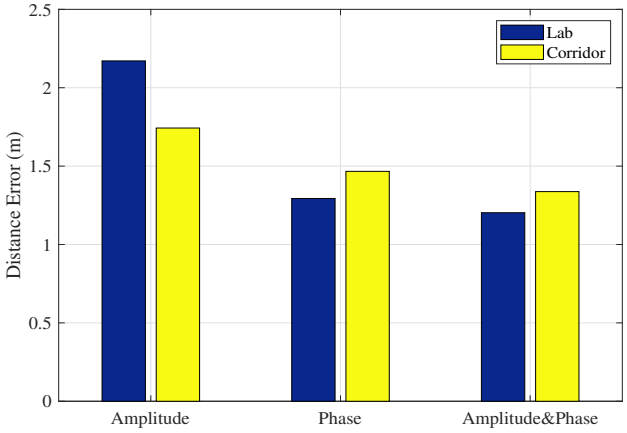


Fig. 16. The average distance errors for different input datasets with the single channel ResLoc model.

Considering that the training process is a one-time procedure in the offline stage, a slightly longer training time does not compromise user experience in the online state. Thus, we choose the dataset with 130 tensors for every training location because of its lowest distance error.

2) Impact of Bimodality: To evaluate the impact of the proposed bimodal CSI data, we apply the ResLoc model to different types of datasets, including: (i) the amplitude only dataset, (ii) the phase difference (i.e., AoA) only dataset, and (iii) the bimodal dataset. We evaluate the performance of ResLoc when using these three types of datasets in the two indoor environments. It is well-known that CSI amplitude values are susceptible to fading and multipath effect. Thus the performance of the amplitude only dataset is the poorest. The computer lab is a cluttered environment. The furniture, computers, and appliances block most of the LOS paths and generate many multipath components. As shown in Fig. 15, the largest error is achieved by the amplitude only dataset in the lab environment. Comparing to CSI amplitude, CSI phase with the periodical change over the propagation distance is relatively more robust. As Fig. 15 shows, a lower distance error is achieved by the phase difference only dataset. The bimodal tensor dataset achieves the lowest distance errors among the three datasets. Due to the use of bi-modal CSI tensor, the phase difference can be utilized to mitigate the influence of complex indoor environments. The mean distance errors are 1.09m and 1.82m in the lab and corridor scenarios, respectively.

3) Impact of Residual Learning Channels: Recall that we also implement a single channel version ResLoc. In this section, we choose the same number of layers as the dual-channel version, and test it with the three datasets used in Section V-C2. The single channel ResLoc results are presented in Fig. 16, where a similar trend as in the previous dual-channel ResLoc case can be observed, meaning that the complexity of the propagation environment is a dominant factor on the localization performance. However, it can be seen that all the errors shown in Fig. 16 are larger than the corresponding errors shown in Fig. 15. For example, the distance error using the amplitude-only dataset in the lab case is 2.17m, while in Fig. 15 it is 1.92m. The lowest distance error of the single channel ResLoc is 1.20m, which is achieved using the bimodal dataset and is still higher than the best result of the dual-channel ResLoc (i.e., 1.09m). According to Fig. 15 and Fig. 16, it is obvious that the dual-channel model enhances the performance of the ResLoc system.

Furthermore, we also compare the distance errors for different ResLoc versions. The dual-channel model of a 3-3-3-3 scheme includes 4 residual blocks in each channel and each block contains 6 convolutional layers. In other words, a dual-channel 3-3-3-3 scheme includes 24 convolutional layers in each channel. To verify the effect of the proposed *residual sharing* approach, a single channel version ResLoc with 48 convolutional layers is trained. Considering that the dual-channel version can leverage the dual-input, the performance of the 48-layer single-channel version will be evaluated in two ways. First, the original input tensor pairs are divided into sets of left-channel tensors and right-channel tensors. Each set is used to train the single-channel ResLoc. Second, we concatenate the input tensor pairs to form a single dataset, which contains all tensors in the original tensor pairs, to train the single-channel ResLoc. Fig. 17 depicts the distance errors. As we can see, the dual-channel version ResLoc still achieves the lowest distance errors in either the lab case or the single corridor case. Even though the single-channel version

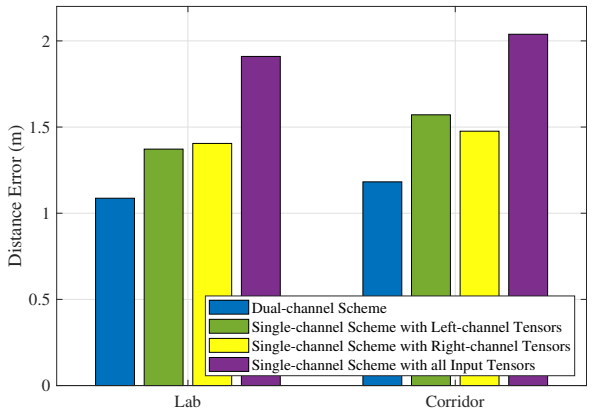


Fig. 17. The average distance errors for single and dual-channel ResLocs with different datasets.

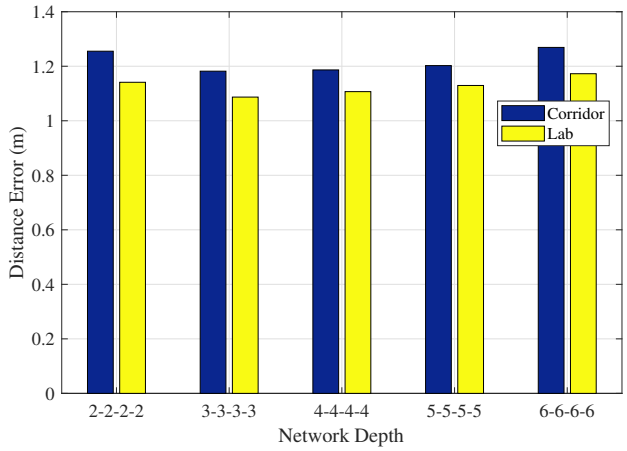


Fig. 18. The average distance errors for different depths of the deep learning network.

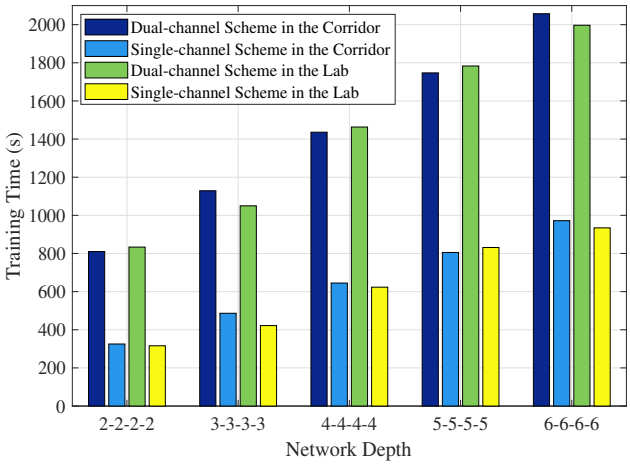


Fig. 19. The average training time for different network depths.

ResLoc has more layers, its distance error is higher than the dual-channel ResLoc. On the contrary, the worst localization performance occurs with the single-channel ResLoc when all tensors are leveraged. It shows that the single-channel ResLoc does not benefit from the ambiguity in the input tensors.

Moreover, because only one slice in each CSI tensor is generated with CSI amplitude, the ResLoc system can be easily upgraded to a triple-channel model. However, the triple-channel model would result in an increase of the number of parameters, which decreases the efficiency of online localization inference. According to Fig. 20, adding a channel could double the testing time. To guarantee the performance of ResLoc system in real-time, the triple-channel model is not used for the current system.

4) Impact of the Number of Layers: We now examine the impact of the number of layers on the ResLoc performance. All convolutional kernels in the residual blocks have an identical size of 3×3. There are four sizes of residual blocks. For each convolutional layer, the number of feature maps in the first block, the second block, the third block, and the forth block are 64, 128, 256, and 512, respectively. Moreover, two convolutional layers are stacked up to form a basic residual block (see Fig. 3). To evaluate how the depth of the network affects its performance, four basic residual blocks are repeated twice, three times, four times, five times, and six times. Theoretically, increasing the number of layers may reduce the distance error. However, Fig. 18 shows that the distance error reaches the lowest point when the ResLoc design is 3-3-3-3 (see Fig. 5). Beyond that the distance error rises slightly as the network gets deeper. We believe that all the schemes are sufficiently deep for the indoor fingerprinting problem. All the distance errors are around 1.2m, which means the distance error is quite robust to the depth of the model. We choose 3-3-3-3 scheme to train the network, because of the lowest distance error and its relatively simpler design.

Obviously, the different numbers of layers affect the training time and testing time. To guarantee fairness in this experiment, all the models are trained with the same configuration. The batchsize and the number of epochs are set to 10 and 50, respectively. The size of CSI tensors is 30×30×3 and 130 tensors in each location are used to train the models. Fig. 19 presents the training times for the single- and dual-channel ResLoc (having the same number of layers) in the corridor and lab scenarios. With the increase of network depth, the training times are gradually increasing. For comparison of the training times between the corridor and lab cases, we find that the training time is not significantly influenced by the environment. However, the depth of the network and the model parameters largely decide the training time. Furthermore, the training time of the dual-channel model is more than twice of the training time of the single channel model. This is because compared to the single channel model, twice amount of weights need to be trained in the dual-channel model, and the dual-channel model takes more time in channel merging.

Similar to the effect of network depth on the training time, the testing time is also affected by the depth of the network. According to Fig. 20, the maximum testing time is 3.1s, which is for the network with depth 6-6-6-6. For the dual-channel model, the time consumption of testing is longer than 1.5s. Because of the complexity of the network, the testing time of the single channel model is shorter, and the maximum testing time of the single channel model is shorter than 2s (note that both Figs. 19 and 20 are obtained with a different GPU (i.e., the Nvidia RTX2080 GPU) from that used for the

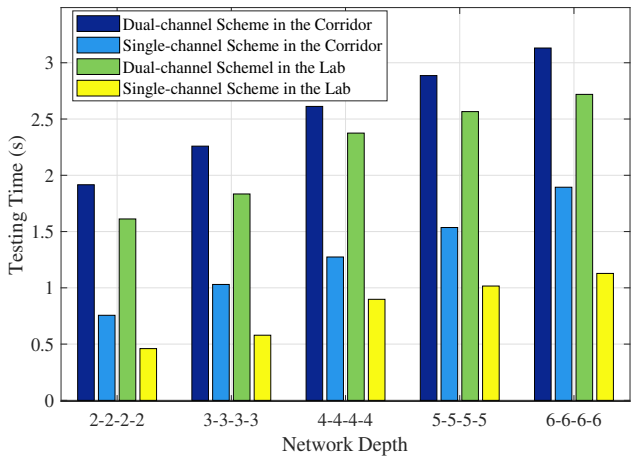


Fig. 20. The average testing time for different network depths.

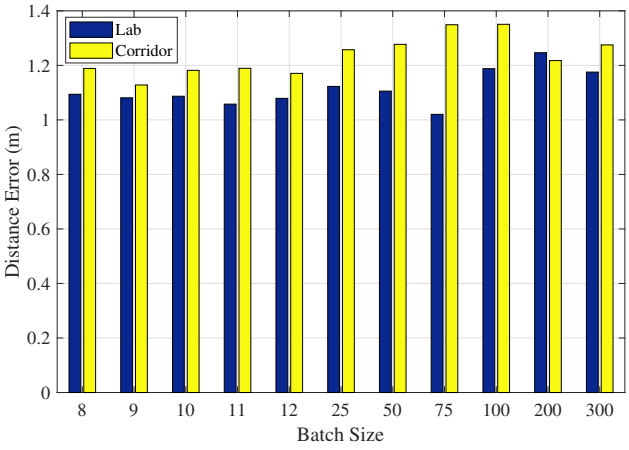


Fig. 21. The average distance errors for different batch sizes.

other figures). Considering the effect of network depth on the distance error, the network design of 3-3-3-3 is the best one, which achieves the lowest distance error with a short training time, and provides location estimation in reasonably short time.

5) Impact of Batch Size: Batch size defines the number of samples that can be propagated through the network. We study the impact of batch size on localization accuracy under the lab and single corridor environments. Fig. 21 presents the mean distance errors for increased batch size from 8 to 300. In the lab case, the largest mean error occurs when the batch size is 200, which is 1.24m. The smallest mean error occurs when the batch size is 75, which is 1.02m. Similarly, In the corridor case, the maximum and minimum errors are 1.36m when the batch size is 100 and 1.11m when the batch size is 9.

6) Impact of Batch Epoch: To improve the accuracy of ResLoc, we also adjust the value of epochs and examine the impact of early stopping on localization accuracy. The results are shown in Fig. 22. In both indoor environments, the largest distance error occurs when epoch is 30. Along with the increase of epoch, the distance error keeps on decreasing. And it remain around 1.1m beyond 50 epochs. Intuitively, the network does not converge before 50 epochs. When the training process convergences, the distance error remains at the

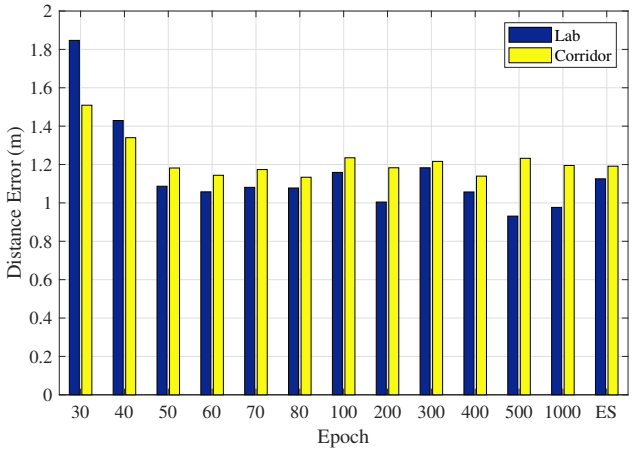


Fig. 22. The average distance errors for different epochs (ES represents early stopping).

same level. On the other hand, we also use early stopping to avoid over-fitting and under-fitting. In Fig. 22, 50 epochs seem to be a good choice to balance training time and localization performance. Thus, 50 epochs are used in the ResLoc system.

7) Measuring Generalization and Overfitting: To evaluate our model's ability to generalize, we shuffled and split our original training dataset into 10 groups to perform 10-fold cross-validation. The results are presented in Table I for the lab case and Table II for the corridor case. Since the cross-validation is repeated 5 times to eliminate randomness, all the results in the tables are averaged across all the 5 trials. As shown in Table I, ResLoc is well trained for all the trials with 100% training accuracy. All the verification accuracy remains around 90%. The average training accuracy is not much higher than the average verification accuracy. Thus, it is safe to say that overfitting is not encountered in the lab experiment. We also calculate the mean distance error with our original testing dataset in the cross-validation. It is obvious that the mean distance errors remain around 1.1m in Table I. Clearly, the mean distance errors in the cross-validation are on the same level with our previous experiment.

The 10-fold cross-validation is also conducted with the corridor dataset. The similarity between each training position is much higher in the corridor case than that in the lab case, which affects the training accuracy as given in Table II. Even though the training accuracy is not as perfect as that in the lab case, it still remains around 98%, which is sufficiently high. The verification accuracy remains around 93% in the corridor scenario as well, which also demonstrates the generalization of our system. The mean distance errors are listed at the bottom of Table II. The overall mean distance error for 5 trials is about 1.16m, which also validates that the errors are on the same level as in our previous experiments.

Finally, we carry out a random test with an extended lab dataset. Specifically, we merge our original training dataset and the original test dataset into a new dataset. The new dataset is composed of the data collected from all the 30 locations in the lab (which are marked in Fig. 6 in both red squares and green dots). This will be our new training dataset. We then build the new test dataset with the CSI data

TABLE I
RESULTS FOR 10-FOLD CROSS-VALIDATION WITH THE LAB DATASET

	fold-1	fold-2	fold-3	fold-4	fold-5	fold-6	fold-7	fold-8	fold-9	fold-10
Training accuracy	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Training loss	0.475	0.531	0.519	0.529	0.499	0.433	0.465	0.602	0.522	0.460
Verification accuracy	0.933	0.901	0.938	0.912	0.913	0.891	0.918	0.899	0.904	0.937
Verification loss	0.706	0.944	0.737	0.926	0.946	0.936	0.752	1.180	0.946	0.687
Mean error (m)	1.111	1.075	1.103	1.081	1.117	1.086	1.116	1.096	1.046	1.098

TABLE II
RESULTS FOR 10-FOLD CROSS-VALIDATION WITH THE CORRIDOR DATASET

	fold-1	fold-2	fold-3	fold-4	fold-5	fold-6	fold-7	fold-8	fold-9	fold-10
Training accuracy	0.976	0.984	0.984	0.986	0.982	0.980	0.979	0.991	0.991	0.985
Training loss	0.546	0.494	0.509	0.491	0.506	0.514	0.528	0.481	0.471	0.518
Verification accuracy	0.944	0.904	0.947	0.922	0.910	0.914	0.928	0.921	0.914	0.926
Verification loss	0.666	0.880	0.630	0.696	0.843	0.844	0.718	0.943	0.792	0.793
Mean error (m)	1.124	1.096	1.190	1.080	1.079	1.208	1.259	1.229	1.215	1.155

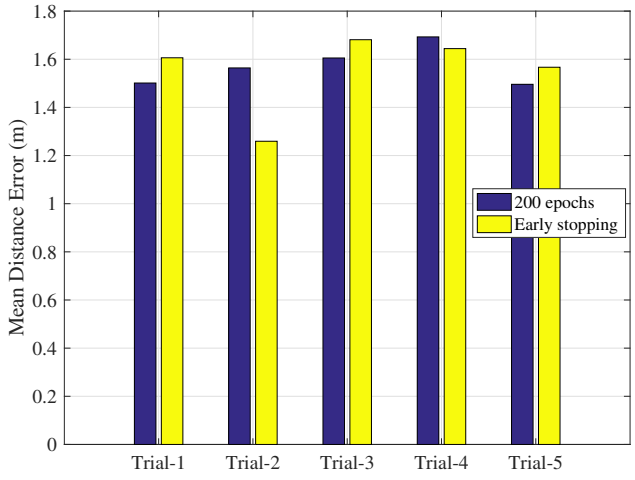


Fig. 23. The average distance errors for randomly selected test locations.

collected from 9 random locations in the lab (different from the 30 marked locations). This way, the ResLoc system is trained for 5 times with different training data and tested at the corresponding random positions. Fig. 23 depicts the mean distance errors for these 5 trials. The results show that the overall mean distance error is 1.57m when ResLoc is trained with 200 epochs. Furthermore, we also utilize early stopping to mitigate the effect of overfitting and speed up the training process. With early stopping, the overall mean distance error is 1.55m. Even though both test results are degraded by the uneven distribution of the training locations, they also prove that our ResLoc system does not overfit the training data in 200 epochs.

VI. CONCLUSIONS

In this paper, we presented ResLoc, a deep residual sharing learning based system for indoor fingerprinting with bimodal CSI tensor data. The proposed deep residual sharing learning model incorporated a dual-channel structure, which has both the advantages of deep residual learning and the novel residual sharing feature. The proposed deep learning model was analyzed with its forward and back. The extensive experimental study validated the superior performance of the proposed ResLoc system.

REFERENCES

[1] 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.

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

[3] D. Li, B. Zhang, and C. Li, "A feature-scaling-based k-nearest neighbor algorithm for indoor positioning systems," *IEEE Internet of Things J.*, vol. 3, no. 4, pp. 590–597, Oct. 2015.

[4] 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.

[5] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the Mona Lisa: Spot localization using PHY layer information," in *Proc. ACM MobiSys'12*, Low Wood Bay, UK, Jun. 2012, pp. 183–196.

[6] D. Halperin, W. J. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *Proc. ACM SIGCOMM'10*, New Delhi, India, Sept. 2010, pp. 159–170.

[7] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity WiFi," in *Proc. ACM Mobicom'15*, Paris, France, Sept. 2015, pp. 53–64.

[8] 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.

[9] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[10] 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.

[11] 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.

[12] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inform. Proc. Syst. 2012*, Lake Tahoe, NV, Dec. 2012, pp. 1106–1114.

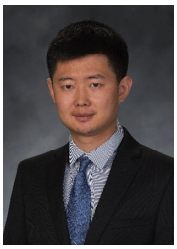
[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR'16*, Las Vegas, NV, June 2016, pp. 770–778.

[14] 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.

[15] 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.

[16] 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*, in press, 10.1109/JIOT.2020.2996564.
[17] B. Harshanand and A. K. Sangaiah, "Comprehensive analysis of deep learning methodology in classification of leukocytes and enhancement using swish activation units," *Springer Mobile Networks and Applications*, pp. 1–19, July 2020.
[18] A. K. Sangaiah, A. Goli, E. B. Tirkolaee, M. Ranjbar-Bourani, H. M. Pandey, and W. Zhang, "Big data-driven cognitive computing system for optimization of social media analytics," *IEEE Access J.*, vol. 8, pp. 82 215–82 226, Apr. 2020.
[19] 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.
[20] 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.
[21] 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.
[22] 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.
[23] 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.
[24] 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.
[25] A. Mittal, S. Tiku, and S. Pasricha, "Adapting convolutional neural networks for indoor localization with smart mobile devices," in *Proc. ACM GLSVLSI' 18*, Chicago, IL, May 2018, pp. 117–122.
[26] T. Zhang and M. Yi, "The enhancement of WiFi fingerprint positioning using convolutional neural network," in *Proc. 2018 International Conference on Computer, Communication and Network Technology*, Wuzhen, China, June 2018, pp. 479–483.
[27] 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. 01 044–01 049.
[28] A. Shokry, M. Torki, and M. Youssef, "DeepLoc: A ubiquitous accurate and low-overhead outdoor cellular localization system," in *Proc. ACM SIGSPATIAL'18*, Seattle, WA, Nov. 2018, pp. 339–348.
[29] H. Rizk, M. Torki, and M. Youssef, "CellinDeep: Robust and accurate cellular-based indoor localization via deep learning," *IEEE Sensors Journal*, vol. 19, no. 6, pp. 2305–2312, Mar. 2018.
[30] Q. Li, H. Qu, Z. Liu, N. Zhou, W. Sun, and J. Li, "AF-DCGAN: Amplitude feature deep convolutional GAN for fingerprint construction in indoor localization system," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–13, Nov. 2019, early access.
[31] H. Rizk, A. Shokry, and M. Youssef, "Effectiveness of data augmentation in cellular-based localization using deep learning," in *IEEE WCNC 2019*, Marrakech, Morocco, Apr. 2019, pp. 1–6.
[32] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proc. ACM NSDI'13*, Lombard. IL, Apr. 2013, pp. 71–84.
[33] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, Mar. 1986.
[34] S. Sen, J. Lee, K. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding WiFi localization," in *Proc. ACM MobiSys'13*, Taipei, Taiwan, June 2013, pp. 249–262.
[35] 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.
[36] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "WiDeo: Fine-grained device-free motion tracing using RF backscatter," in *Proc. ACM NSDI'15*, Oakland, CA, May 2015, pp. 189–204.
[37] W. Gong and J. Liu, "Robust indoor wireless localization using sparse recovery," in *Proc. IEEE ICDCS'17*, Atlanta, GA, June 2017, pp. 847–856.
[38] X. Li, D. Zhang, Q. Lv, J. Xiong, S. Li, Y. Zhang, and H. Mei, "IndoTrack: Device-free indoor human tracking with commodity Wi-Fi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. Article No.: 72, Sept. 2017.
[39] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proc. ACM MobiCom'14*, Maui, Hawaii, Sept. 2014, pp. 483–494.
[40] 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.
[41] C. R. Karanam, B. Korany, and Y. Mostofi, "Magnitude-based angle-of-arrival estimation, localization, and target tracking," in *Proc. IEEE IPSN 2018*, Porto, Portual, Apr. 2018, pp. 254–265.
[42] A. Ahmed, R. Arablouei, H. Hoog, B. Kusy, R. Jurdak, and N. Bergmann, "Estimating angle-of-arrival and time-of-flight for multi-path components using WiFi channel state information," *MDPI Sensors J.*, vol. 18, no. 6, p. 1753, 2018.
[43] 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.
[44] 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.
[45] 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.
[46] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
[47] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. ACM ICML'10*, Haifa, Israel, June 2010, pp. 807–814.
[48] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. ACM OSDI'16*, Savanna, GA, Nov. 2016, pp. 265–283.

Xiangyu Wang received a B.S. degree in electrical engineering from Taiyuan Institute of Technology, Taiyuan, China in 2014, and an MS degree in Electrical and Computer Engineering (ECE) from Auburn University, Auburn, AL in 2017. He has been pursuing a PhD in ECE at Auburn University since Spring 2018. His research interests focus on machine learning, indoor localization, and IoT. He is a co-recipient of the Best Student Paper Award of IEEE PIMRC 2017.



Xuyu Wang [S'13-M'18] received the M.S. in Signal and Information Processing in 2012 and B.S. in Electronic Information Engineering in 2009, both from Xidian University, Xi'an, China. He received a Ph.D. in Electrical and Computer Engineering from Auburn University, Auburn, AL, USA in Aug. 2018. He is an Assistant Professor in the Department of Computer Science, California State University, Sacramento, CA. His research interests include indoor localization, deep learning, and big data. He is a co-recipient of the Second Prize of Natural Scientific Award of Ministry of Education, China in 2013, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the Best Paper Award of IEEE GLOBECOM 2019, The 2018 Best Journal Paper Award of IEEE Communications Society Multimedia Communications Technical Committee, the Best Demo Award of IEEE SECON 2017, and the Best Student Paper Award of IEEE PIMRC 2017.

Shiwen Mao [S'99-M'04-SM'09-F'19] received his Ph.D. in electrical engineering from Polytechnic University, Brooklyn, NY (now New York University Tandon School of Engineering) in 2004. He was a postdoc at Virginia Tech since 2004 and joined Auburn University, Auburn, AL in 2006 as an Assistant Profess of Electrical and Computer Engineering. He held the McWane Endowed Professorship from 2012 to 2015 and the Samuel Ginn Endowed Professorship from 2015 to 2020. Currently, he is a Professor and the Earle C. Williams Eminent Scholar, and the Director of the Wireless Engineering Research and Education Center (WEREC) at Auburn University.

His research interest includes wireless networks, multimedia communications, and smart grid. He is a Distinguished Speaker (2018-2021) and was a Distinguished Lecturer (2014-2018) of the IEEE Vehicular Technology Society. He was the chair of IEEE ComSoc Multimedia Communications Technical Committee (MMTC) for 2016-2018. He was the TPC Co-Chair of IEEE INFOCOM 2018 and will be the TPC Vice-Chair of IEEE GLOBECOM 2022. He is an Associate Editor-in-Chief of IEEE/CIC China Communications, an Area Editor of IEEE Transactions on Wireless Communications, IEEE Open Journal of the Communications Society, IEEE Internet of Things Journal, and ACM GetMobile, and an Associate Editor of IEEE Transactions on Network Science and Engineering, IEEE Transactions on Mobile Computing, IEEE Multimedia, and IEEE Networking Letters, among others.

He received the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award in 2019, the IEEE ComSoc MMTC Distinguished Service Award in 2019, Auburn University Creative Research & Scholarship Award in 2018, the 2017 IEEE ComSoc ITC Outstanding Service Award, the 2015 IEEE ComSoc TC-CSR Distinguished Service Award, the 2013 IEEE ComSoc MMTC Outstanding Leadership Award, and NSF CAREER Award in 2010. He is a co-recipient of the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the 2018 IEEE ComSoc MMTC Best Journal Paper Award, the 2017 IEEE ComSoc MMTC Best Conference Paper Award, the Best Demo Award of IEEE SECON 2017, the Best Paper Awards of IEEE GLOBECOM 2019, 2016, & 2015, IEEE WCNC 2015, and IEEE ICC 2013, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.