freeLoc: Wireless-based Cross-Domain Device-free Fingerprints Localization to free User's Motions

Dawei Yan, Fei Shang, Panlong Yang, *Member, IEEE*, Feiyu Han, Yubo Yan, *Member, IEEE*,
and Xiang-Yang Li *Fellow, IEEE*

***Abstract*— Due to contactless and convenient experiences, WiFi-based device-free fingerprints localization technologies have extensively attracted research attention. However, they are studied based on an assumption that the user is stationary and face a major challenge in the presence of users' motions. That is because users' motions induced CSI WiFi variations results in inconsistent location fingerprints during training and prediction, leading to system ineffective. To solve this problem, in this paper, we propose *freeLoc*, which aims to free users' motions (even unseen) while maintaining accurate localization. Specifically, we construct a domain adaptation network that defines different users and motions as different domains, and learns domain-independent representations to extract location fingerprints independent of users' motions. Unfortunately, collecting sufficient amounts of WiFi data is difficult. To reduce the cost of labeling data and ensure the performance of domain adaptation network, we utilize adversarial autoencoder to build a data augmentation module to introduce data diversity. We deploy experiments in a real scenario, and the results show that only by labeling three motions of three users, we can achieve accurate localization (the nearest locations are about one meter away) for a total of 36 domains including 6 users and 6 motions. Compared to other existing technologies, *freeLoc* can improve location prediction accuracy by up to 35%.**

***Index Terms*—Device-free localization, fingerprints inconsistency, domain adaptation, data augmentation.**

I. Introduction

Stable and accurate *indoor location based service* (ILBS) can provide key opportunities for many applications. For example, by pushing location updates to users, *Internet of Things* (IoT) applications such as geo-social networking [1], *point-of-interest* (POI) recommendations [2], and smart home automation [3] can be implemented. More accurate location awareness can provide solutions for applications such as unmanned supermarkets [4], *augmented reality* (AR) [5]. In recent years, many wireless indoor localization technologies have been extensively studied, such as WiFi [6]–[10], RFID [11], [12], acoustics [13], and vision [14], [15]. Among them, WiFi has gradually become one of the favored solutions due to its ubiquitous infrastructures and low cost.

WiFi-based localization systems are mainly divided into device-oriented and device-free solutions. The device-oriented approaches achieve decimeter-level localization accuracy by measuring the *angle of arrival* (AoA) [8], [9], *time of flight* (ToF) [16] or *received signal strength indicator* (RSSI) [17] from the smartphone or other WiFi device carried by the user to the WiFi *access point* (AP). Due to the weakened requirement for users to carry smart devices, device-free localization systems can be applied in a wider range of scenarios. Generally, device-free localization system associates the collected RSSI or *channel state information* (CSI) with the location of the user, and then extracts the location fingerprints and trains the neural network to achieve decimeter-level localization accuracy [18]–[20]. In particular, CSI has rich frequency and space diversity information, and its magnitude is more stable in time series, so CSI fingerprint is more representative in terms of location features.

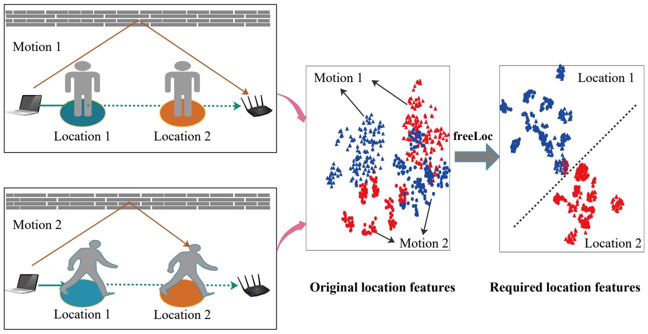


Fig. 1. Users are often accompanied by different motions when being located, which leads to inconsistency of location fingerprints and makes the device-free localization systems ineffective. *freeLoc* does not limit the users' motions, while remaining accurate localization.

Although the device-free fingerprints localization technologies provide some convenience for users, the existing arts still have many deficiencies in the application of real-world scenarios. It is known that CSI is extremely sensitive to the surrounding environment for it fine-grained reflects channel characteristics. Previous arts have studied the impact of changes in environment and users on CSI location fingerprints [21], [22]. However, unlike them, we observe that when the localization systems are working, users do not always remain standing in a certain location, but may be accompanied by various motions, such as extending arms, bending waist, walking with arms swinging and etc. These motions change WiFi propagation characteristics, even for the same user in the same location. What's more, this inconsistency from motions is more common than changes in the environment and users, and limiting the user's motions makes the localization systems less convenient and practical. Therefore, in this paper, we focus

D. Yan, F. Shang, F. Han, Y. Yan and X.-Y Li are with CAS Key Laboratory of Wireless-Optical Communications, University of Science and Technology of China, Hefei 230021, China.(e-mail: yandw@mail.ustc.edu.cn, shf_1998@outlook.com, fyhan@mail.ustc.edu.cn, yuboyan@ustc.edu.cn, xiangyangli@ustc.edu.cn).P. Yang is with Nanjing University of Information Science & Technology, Nanjing 210044, China.(e-mail: plyang@ustc.edu.cn).
D. Yan and F. Shang contributed equally to this work.

on reducing the constraints of motions to make users of the localization system more relaxed, while remaining accurate.

As mentioned above, after data training on specific motions of specific groups of people, different motions change the WiFi transmission characteristics. If the fingerprints are not re-collected to train the localization systems, these may lead to poor localization results. Fig. 1 shows a simple example. It can be clearly seen that the WiFi reflection path between the WiFi router and the laptop changes due to the difference in motions. As a result, even if the same person stands at the same location, the location features of the two situations are very different, eventually causing the features of different locations to overlap together. Therefore, if localization system only train on one condition, it may be bad to predict the localization results of the other condition. Detailed analysis of CSI fingerprints inconsistency and **observation** are given in Section II-B.

In theory, if we are able to collect CSI fingerprints from all possible motions at each location, we could train a localization system that is immune to the diversity of motions. However, considering the richness of motions and the large differences in motions of people with different body sizes, this will be extremely time-consuming and labor-intensive, so this solution is not practical in real world. Another easy-to-think solution is that we mark the CSI when each user has a new motion, and then update the fingerprint library of the localization system in real time, so that the location fingerprint includes all known motions. Unfortunately, this requires accurate recognition of each motion, even for different locations and different people, which currently no motion recognition system can achieve this.

Therefore, in order to solve the problem of inconsistent WiFi CSI fingerprints caused by different motions, we must solve the following two main challenges:

- **Challenge 1:** different motions have different impacts on WiFi CSI fingerprints. In order to eliminate these influences and ensure localization accuracy, we need to cover all possible motions when creating and training fingerprints. However, it is indeed practically impossible to cover all possible motions.
- **Challenge 2:** different users have different motions due to their different body shapes and habits. In order to eliminate these influences, even for the same motion, we still need to cover all possible users. However, it is impractical to label all users' data associated with the corresponding locations.

To tackle above challenges, we propose *freeLoc*, a WiFi device-free localization system based on a fine-grained deep learning framework of domain adaptation and data augmentation, which is capable of locating different users accompanied by different motions, only collecting and labeling a small amount of CSI fingerprints.

Firstly, we design a localization system based on the domain adaptation technology [23] to reduce the coupling of fingerprints features and users' motions. When the user's motion changes, *freeLoc* can predict the location well without relabeling the data. Specifically, *freeLoc* regards the users' motions with location labels in the original fingerprints database as the source domains, and the emerging users' motions without location labels as the target domains. Then, we train the model using all the data from the source and target domains together. The ultimate goal is to learn the common features of all data in the source and target domains, while downplaying the differences between the source and target domains. Therefore, we pass a domain discriminator to identify different domains, and the feature extractor is designed to try to fool the domain discriminator to minimize the prediction accuracy of the domain discriminator, while maximizing the location prediction accuracy together with the location predictor. In this way, *freeLoc* can predict different data from different domains (*i.e.*, different motions) without relabeling. In addition, in order to improve the accuracy of location prediction, we carefully design the loss function. Specifically, we introduce loss functions with labeled and unlabeled data respectively in location predictor, and input the data with location prediction results and without location prediction results to the domain discriminator respectively.

Then, to obtain sufficient source domains' data to train the domain adaptive network, we generate synthetic data similar to the collected location-labeled data based on the data augmentation model. In this way, we can expand the scale of source domains' data, and increase the diversity of users and motions in the source domains' data, so that a very rich set of users and motions can be covered with a small amount of labeled data. Specifically, we use one *adversarial autoencoder* (AAE) [24] for a small number of user fingerprints for each location, and then separately generate fingerprints for each location that are different from the base users and motions. It is worth noting that these synthetic fingerprints are also location-labeled. Therefore, *freeLoc* can predict data about more users and motions without collecting and labeling their fingerprints.

Overall, the contributions are summarized as follows:

- In this paper, we analyze the impact of different users' motions on WiFi-based device-free fingerprints localization. Then, we proposed *freeLoc*, which aims to free users' diverse motions (even unseen) while maintaining accurate localization.
- We use domain adaptation technology to reduce the coupling of location fingerprints with users and motions, *i.e.*, we treat different users and motions as different domains and enable *freeLoc* to learn domain-independent location features. We then use data augmentation methods such that using only a small amount of location-labeled data, *freeLoc* can handle well a variety of rich, location-unlabeled data for different users and motions.
- We build a localization testbed based on four ASUS RT-AC86U routers with Nexmon CSI extractors installed, and evaluate *freeLoc* in an empty room of $8m \times 10m$. We define six different motions and invite six volunteers of different heights and weights to participate in the experiments. We ask each volunteer to perform each motion at ten different locations (the closest distance is about one meter). The results show that *freeLoc* only by labeling three motions of three users, accurate localization of all 36 domains can be achieved, which is at least 12% better than other existing technologies (the worst-case accuracy by 35%).

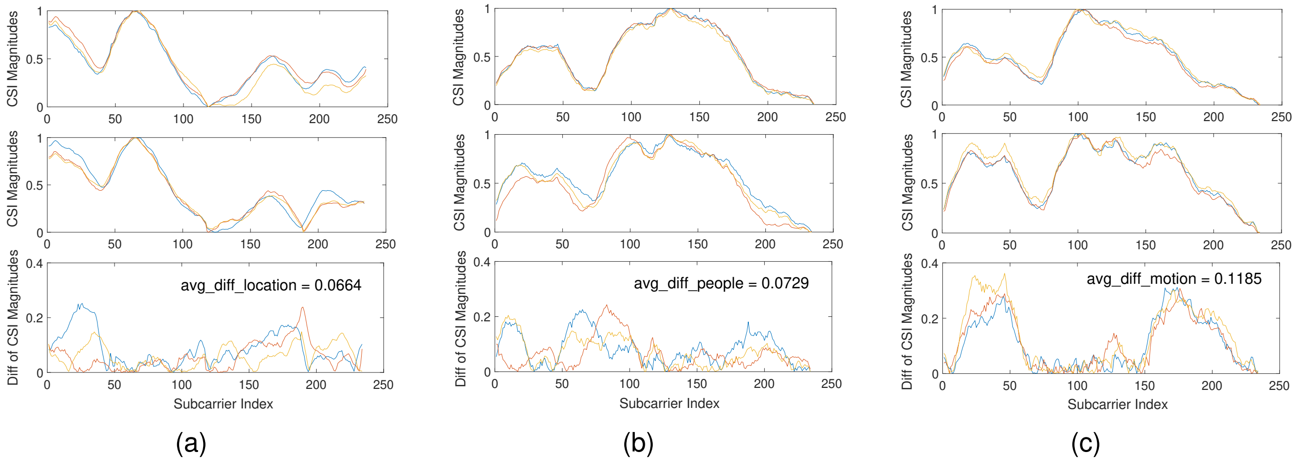


Fig. 2. **WiFi CSI magnitudes are inconsistent due to different locations, different people, or different motions, and the impact of motion may be more important than location and people:** (a) Same people with the same motion at different locations lead to different WiFi CSI magnitudes. (b) Different people with the same motion at the same location lead to different WiFi CSI magnitudes. (c) Different motions of the same people at the same location lead to different WiFi CSI magnitudes.

The rest of this paper is organized as follows: Section II presents the preliminary tests of WiFi CSI fingerprints inconsistency. In Section III, we provide an overview of *freeLoc*. In Section IV, we describe the modules design of proposed framework in details. Implementation and evaluation are presented in Section V and Section VI. Section VII discusses the related works. We finally conclude our work in Section VIII.

II. CSI Fingerprints and Inconsistency

In this section, we present the basics of device-free localization based on WiFi CSI fingerprints, give the general form of input data using WiFi CSI as fingerprints, and analysis the inconsistencies in CSI fingerprints caused by different locations, different people and different motions based on tests in a real environment.

A. Passive Localization based on CSI Fingerprints

In wireless communication, *channel state information* (CSI) is the channel attribute of the communication link. It describes the attenuation factor of the signal on each transmission path, that is, the value of each element in the channel matrix $\mathbf{H}$, such as signal scattering, multipath, shadowing fading, power decay of distance and etc [25]. Therefore, for fixed WiFi Tx and Rx, when people stand in different locations, they will absorb or reflect WiFi signals in different ways, making the value of CSI different. A passive localization system marks these different CSI values, thereby enabling an estimate of the person's location [22]. In addition, some tools such as *Intel 5300 CSI Tool*, *Atheros CSI Tool* and *Nexmon CSI Extractor* have been able to extract CSI from received data packets and parse out the channel matrix $\mathbf{H}$.

Recently, WiFi technologies commonly use *orthogonal frequency division multiplexing* (OFDM) and *multi-input multi-output* (MIMO), thus channel between each antenna pair of Tx-Rx consists of multiple subcarriers. Assume that Tx-Rx has a total of $N_{ss} = N_{tx} \times N_{rx}$ spatial streams, and each WiFi channel is divided into N_{sc} frequency subcarriers by OFDM. In this paper, we use the amplitudes of CSI subcarriers on all spatial streams at time t as data input to the localization system, *i.e.*, $\mathbf{X}[t]$ [22]:

$$\mathbf{x}[t] = \left(\left|h_{1,1}[t]\right|, \cdots, \left|h_{1,N_{sc}}[t]\right|, \cdots, \left|h_{N_{ss},N_{sc}}[t]\right|\right), \quad (1)$$

where $h_{i,j}[t]$ represents the CSI value of the i-th stream on the j-th subcarrier collected at time t, and $|\cdot|$ denotes the the magnitudes of complex numbers.

B. CSI Fingerprints Inconsistency Analysis

To verify the CSI fingerprints inconsistency caused by the motions of people, we perform preliminary tests in an empty room of $8m \times 10m$. Specifically, we ask four volunteers of different body sizes to perform three motions (*i.e.*, keeping standing, stretching arms, bending waist) at five different locations in the room. Then we collect the CSI of each condition based on ASUS RT-AC86U router and Nexmon [26] at the ISM band 5GHZ/80MHz, and the duration of each condition is one minute. During the collection of CSI, we make the volunteers fix a certain motion. It is worth noting that there are no changes in the environment and WiFi transceivers during the collection of CSI for controlling variables. We extract the magnitudes of the subcarriers of the first spatial stream of each CSI packet (removing the subcarriers used for protection and null), and perform normalization processing to specifically analyze the distribution of CSI magnitudes at different locations, different people and different motions.

We first compare the CSI magnitudes of different locations, where both the person and the motion are the same. As shown in Fig. 2a, the CSI magnitudes of some subcarriers are different at different locations. Then, we compare different people, where both the location and the motion are the same. As shown in Fig. 2b, the CSI magnitudes are not exactly the same of all subcarriers for different people. Finally, we compare the same person with different motions at the same location, and Fig. 2c shows the result. To further analyze the influence from different locations, people and motions,

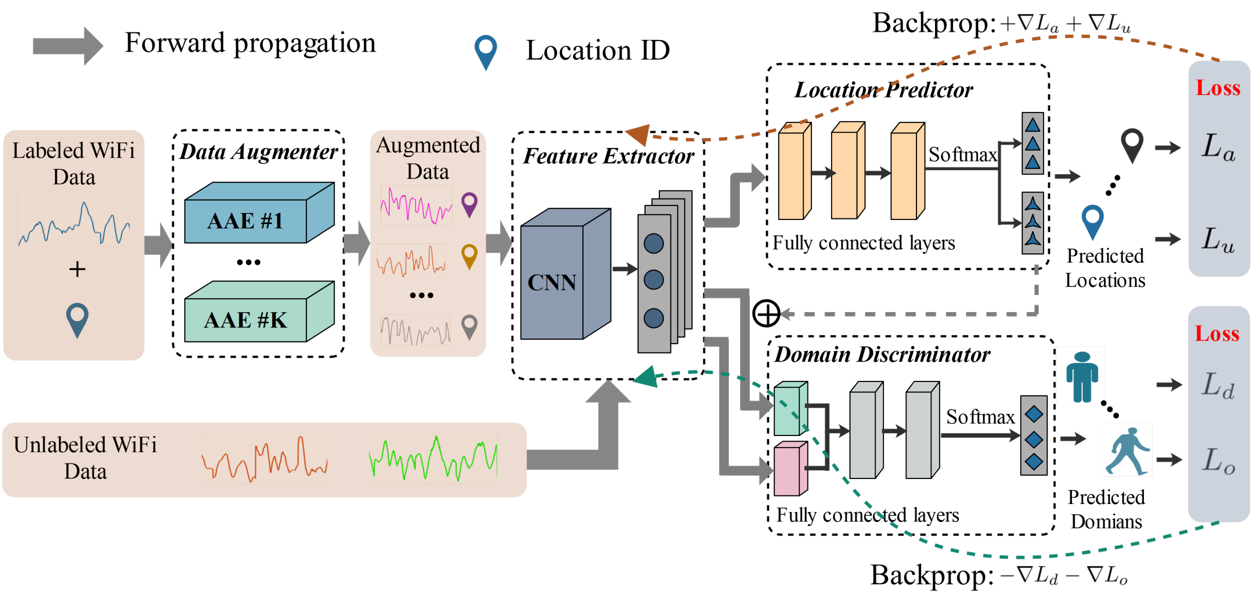


Fig. 3. **System overview:** the location-labeled data pass the **data augmenter** to generate many similar augmented data, together with the unlabeled data go through the **feature extractor** to generate latent features, then we use the **location predictor** to obtain predicted locations that maximize localization accuracy and take positive during backpropagation, while using the **domain discriminator** to minimizing domain prediction accuracy to learn domain-independent features take negative during backpropagation.

we analyze the similarities between them. Specifically, we differ the CSI magnitudes of the three situations and obtain their average values. The results in Fig. 2 show that the difference introduced by different motions is greater than the difference between locations and people. Obviously, we have the observation, *i.e.*, **Observation: even if the same person is in the same location, the difference in CSI magnitudes may be large when the motions are different. As shown in Fig. 2, the impact of motions may be more important than locations and people. The localization accuracy may be poor when users perform untrained motions at the predicted location, as shown in Fig. 1.**

From the above analysis and two key observations, we think that the user's own motions do have an impact on the WiFi CSI location fingerprints, and it is necessary to eliminate this impact. Next, we illustrate how our proposed model does this while improving location prediction accuracy.

III. System Overview

A. Labeled/Unlabeled Data

In this paper, we focus on the impact of the users' own motions on CSI fingerprints localization. We collect CSI data when different people do different motions and extract the magnitudes as fingerprints as shown in Equa. 1. As shown in Fig. 3, for some motions, the data of some people is labeled with location labels, *i.e.*, labeled datasets:

$$\mathbf{X}_L = \{(\mathbf{x}_L[t], \mathbf{y}_L[t])\}, \tag{2}$$

where $\mathbf{x}_L[t]$ is a labeled CSI data, and $\mathbf{y}_L[t]$ is the location label. While for other people or other motions, there are no location labels, *i.e.*, unlabeled datasets $\mathbf{X}_U = \{\mathbf{x}_U[t]\}$. We regard different people and motions as different domains, where the domains with location labels are the source domains, and the other domains are the target domains. We use location-labeled source domains data $\mathbf{D}_S$ for initializing the system and together with unlabeled source or target domains data $\mathbf{D}_T$ for training the predictive model. Our goal is to have good localization results for both location-labeled and location-unlabeled data $\mathbf{X} = \mathbf{X}_L \cup \mathbf{X}_U$.

B. Model Architecture

As shown in Fig. 3, the proposed model mainly consists of four modules: data augmenter, feature extractor, location predictor and domain discriminator.

Data augmenter. Firstly, the location-labeled data enters the data augmenter based on AAE, which generates data separately based on the existing labeled data for each location, to expand more potential users and motions.

Feature extractor. Secondly, We feed the augmented and unlabeled data into the feature extractor based on *convolutional neural network* (CNN) to generate latent features for low-dimensional representations.

Location predictor. Thirdly, based on the generated latent features, the location predictor is used to obtain the predicted location that maximizes the localization accuracy.

Domain discriminator. At the same time, in order to eliminate the impact of different motions (*i.e.*, domain-specific features), we design a domain discriminator to predict each domain, *i.e.*, identify which person do which motion. The goal of the domain discriminator is to maximize domain labeling accuracy, which seems contradictory to learning domain-independent features. However, the feature extractor is designed to try its best to fool the domain discriminator, *i.e.*, minimize its prediction accuracy, while improving the location prediction accuracy. In this way, we achieve learning common motion-independent features for defined locations. Finally, we

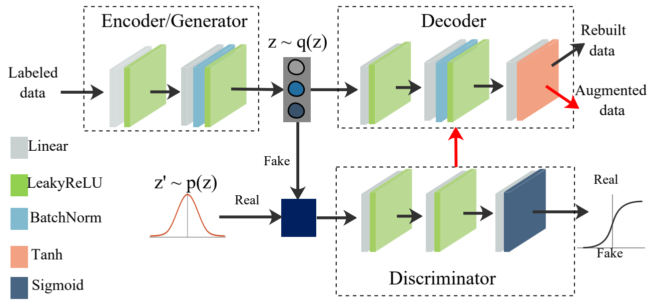


Fig. 4. Data augmenter based on AAE for labeled CSI data.

introduce two constraints to optimize the model for improving location prediction accuracy.

IV. Design

A. Data Augmenter

In order to better eliminate the influence of different motions from different people, as much CSI data as possible from different users and motions should be collected. However, the time and labor cost of collecting and labeling data is huge, so we use data augmentation to generate more fingerprints. Specifically, we design the AAEs shown in Fig. 4, and use one AAE for each location following the location markers of the fingerprints. AAE [24] is a generative model that applies adversarial ideas to the *variational autoencoder* (VAE) training process, and its typical network architecture consists of an *autoencoder* (AE) and a *generative adversarial network* (GAN). VAE often leave regions in the space of prior distributions that do not map to actual samples in the data. AAE aims to improve this situation by encouraging the encoder's output to completely fill the space of the prior distribution, thus enabling the decoder to generate realistic samples from any data point sampled.

During training, the data augmenter divides the labeled data into K subsets $\mathbf{X}_L^k, k = 1, 2, \cdots, K$, each of which corresponds to the fingerprints of one location. Firstly, the labeled data x^k of the kth AAE is fed into the encoder of AAE$\#k$ to generate a latent vector $z \sim q(z)$, where $q(z)$ is aggregated posterior distribution. z is sent to the decoder, and generate a vector $\hat{x}^k$ to reconstruct the labeled data x^k. We define the reconstruction loss L_B^k using *mean square error* (MSE):

$$L_B^k = \frac{1}{2N_G^k} \sum_{i=1}^{N_G^k} \left(x_i^k - \hat{x}_i^k\right)^2, \tag{3}$$

where N_G^k is the number of samples.

Then, we train the discriminator to regularize the rebuild data. At this time, the encoder of AE becomes the generator of GAN, and its output x^k is sent to the discriminator together with the vector z' that obeys the prior distribution $p(z)$. We compare several common prior distributions, such as Gaussian distribution, uniform distribution, and Laplace distribution, and we chose Gaussian distribution $N(0, k\delta^2)$ as the prior distribution $p(z)$. For the discriminator, the label a^k is 0 when x^k is used as input, and the label a^k is 1 when z' is used as input. Cross entropy is used as the loss function L_G^k of the discriminator:

$$L_G^k = -\frac{1}{N_G^k} \sum_{i=1}^{N_G^k} \left(a_i^k \log\left(\hat{a}_i^k\right) + \left(1 - a_i^k\right) \log\left(1 - \hat{a}_i^k\right)\right), \tag{4}$$

The loss function of the whole AAE$\#k$ is defined as the sum of the reconstruction loss and the discriminator loss:

$$L_{AAE}^k = L_B^k + L_G^k. \tag{5}$$

It is worth noting that because we perform data synthesis on the data for each location, we can assign location labels to the data after they have been synthesized for creating labeled synthetic data:

$$\mathbf{X}_S^k = \left\{(\mathbf{x}_S^k)[t], k)\right\}. \tag{6}$$

Our goal for synthetic data is that the size of the synthetic data at the kth location is 20 times larger than the size of the original labeled data. Finally, we collect all the synthetic data $\mathbf{X}_S$ together with the source data as labeled augmented data $\mathbf{X}_A = \mathbf{X}_S \cup \mathbf{X}_L$.

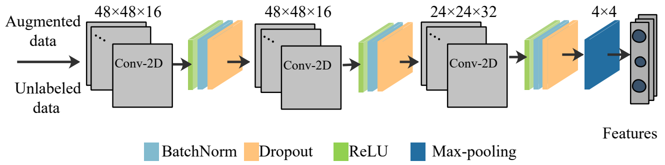


Fig. 5. Feature extractor based on CNN for augmented or unlabeled data.

B. Feature Extractor

The feature extractor is used to extract location-related features to maximize location prediction accuracy while minimizing domain prediction accuracy. We feed all the augmented labeled data and unlabeled data together into the feature extractor to output their feature vectors. In this paper, we use the widely adopted CNN to extract location features. Specifically, as shown in Fig. 5, we employ a three-layer stacked CNN to extract features. At each layer of the CNN, we use convolutional layers with 2D convolutional kernels, utilize *rectified linear units* (ReLU) to introduce nonlinearities, and insert batch normalization layers to speed up training and dropout layers to avoid overfitting. Then, we use a max-pooling layer to reduce the size of the representation. Given the input data $\mathbf{X}_i$, we can obtain features as follows:

$$\mathbf{Z}_i = \mathrm{CNN}(\mathbf{X}_i; \Theta_{cnn}), \tag{7}$$

where Θ_{cnn} is the parameter set of CNN.

C. Location Predictor

After extracting the feature vector $\mathbf{Z}_i$, we feed it into the location predictor to predict the location, as shown in Fig. 3. Firstly, we learn a representation $\mathbf{V}_i$ for $\mathbf{X}_i$ using three fully connected layers and the activation function ReLU:

$$\mathbf{V}_i = \mathrm{ReLU}(\mathbf{Z}_i; \Theta_{fc}), \tag{8}$$

where Θ_{fc} is the parameter set of ReLU. The reason for using three fully connected layers is that more parameters can be learned, and more fully connected layers do not improve performance much. Then to predict the location, we map $\mathbf{V}_i$ to a new latent space $\mathbb{R}^C$, where C is the number of location labels. Finally, we pass $\mathbf{V}_i$ through an output layer whose activation function is softmax, so as to obtain the predicted probability vector of a certain location:

$$\hat{y}_i = \mathrm{Softmax}\left(\boldsymbol{W}_v V_i + b_v\right), \tag{9}$$

where $\boldsymbol{W}_v$ and b_v are the parameters.

To improve the accuracy of location prediction, we predict location labels for labeled and unlabeled data [27], and record them as $\hat{y}_i^l$ and $\hat{y}_i^u$, respectively. For both labeled and unlabeled data, we use cross-entropy as the loss function:

$$L_a = -\frac{1}{N_l}\sum_{i=1}^{N_l}\sum_{c=1}^{C} \mathrm{y}_{ic}^l \log\left(\hat{\mathrm{y}}_{ic}^l\right), \tag{10}$$

$$L_u = -\frac{1}{N_u}\sum_{i=1}^{N_u}\sum_{c=1}^{C} \hat{\mathrm{y}}_{ic}^u \log\left(\hat{\mathrm{y}}_{ic}^u\right), \tag{11}$$

where N_l and N_u are the number of labeled and unlabeled data used for training, respectively. Finally, the overall loss function for the location predictor is as follows:

$$L_{LP} = L_a + \gamma L_u, \tag{12}$$

where γ is the weight value used to constrain the influence of unlabeled data on the decision of the location predictor.

D. Domain Discriminator

Domain adaptation is a special case of transfer learning, the idea is to map data features in different domains to the same feature space, so that domain-invariant features can be obtained [28]. In this paper, we use domain adversarial deep learning [23] techniques to leverage unlabeled data to eliminate features related to the users' motions. In our adversarial network, a domain is defined as a pair of user and motion. The reason for this definition is that different motions have different effects on the WiFi CSI location fingerprints, and different users have different body sizes. Specifically, our goal is to design a domain discriminator to recognize different users and motions, combined with the feature extractor to make it fool the domain discriminator, resulting in location features that are independent of the users' motions.

Firstly, the output $\mathbf{Z}_i$ of the feature extractor contains not only location-specific features, but also domain-specific features already. As shown in Fig. 3, in order to match the conditional distribution, we concatenate $\mathbf{Z}_i$ with the predicted label output $\hat{\mathrm{y}}_i$ of the location predictor [29], which together feed into the domain discriminator to predict the domain label [30]:

$$\mathrm{C}_i = \mathrm{Z}_i \oplus \hat{\mathrm{y}}_i, \tag{13}$$

where $\oplus$ is the concatenation operation. Next, we learn a representation $\mathbf{M}_i$ of $\mathbf{C}_i$ using three fully connected layers and the activation function ReLU:

$$\mathbf{M}_i = \mathrm{ReLU}(\mathbf{C}_i; \Theta_{dd}), \tag{14}$$

where Θ_{dd} is the parameter set of ReLU. Then, to predict the domain, we map $\mathbf{M}_i$ to the domain distribution space and pass $\mathbf{M}_i$ through an output layer whose activation function is softmax, so as to obtain the predicted probability vector of a certain domain:

$$\hat{d}_i = \mathrm{Softmax}\left(\boldsymbol{W}_m M_i + b_m\right), \tag{15}$$

where $\boldsymbol{W}_m$ and b_m are the parameters.

Algorithm 1 Model Training Process.

```
Input: Augmented data $\mathbf{X}_A$ with labels $\mathbf{Y}_A$, unlabeled data
    $\mathbf{X}_U$, location number $C$, domain number $D$.
Output: Model with parameters $\Theta_{cnn}$, $\Theta_{fc}$, $\Theta_{dd}$, $(\mathbf{W}_v, b_v)$
    and $(\mathbf{W}_m, b_m)$.
Initialize parameters $\Theta_{cnn}, \Theta_{fc}, \Theta_{dd}, (\mathbf{W}_v, b_v), (\mathbf{W}_m, b_m)$;
for $i \leftarrow 1; i <= C; i++$ do
    while not done do
        Sample data $\mathbf{X}$ with $\mathbf{Y}$ from $\mathbf{X}_A$ with $\mathbf{Y}_A$ and $\mathbf{X}_U$;
        Obtain $\mathbf{Z}$ via passing $\mathbf{X}$ by Equ. 7;
        Obtain $\mathbf{V}$ via passing $\mathbf{Z}$ by Equ. 8;
        predict location labels $\hat{\mathbf{y}}$ via passing $\mathbf{V}$ by Equ. 9;
        Compute location predictor loss $L_{LP}$ using Equ. 12;
        Predict domain labels $\hat{\mathbf{d}}^o$ via passing by Equ. 17;
        Obtain $\mathbf{C}$ via passing $\mathbf{Z}$ and $\hat{\mathbf{y}}$ by Equ. 13;
        Obtain $\mathbf{M}$ via passing $\mathbf{C}$ by Equ. 14;
        Predict domain labels $\hat{\mathbf{d}}$ via passing by Equ. 15;
        Compute domain discriminator loss $L_{DD}$ using
        Equ. 19;
    end while
end for
Obtain overall loss $L_{all}$ by Equ. 20;
Update parameters $\Theta_{cnn}, \Theta_{fc}, \Theta_{dd}, (\mathbf{W}_v, b_v), (\mathbf{W}_m, b_m)$.
```

We use cross-entropy as the loss function for domain label prediction:

$$L_d = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{D} \mathbf{d}_{ij} \log\left(\hat{\mathbf{d}}_{ij}\right) \tag{16}$$

where N is the number of training data, and D is the number of domains. Furthermore, to improve the performance of the domain discriminator, we not only align the posterior distributions of the source and target domains, but also align the marginal distributions between the source and target domains. Specifically, we directly feed the output $\mathbf{Z}_i$ of the feature extractor into the domain discriminator, and go through the same process to predict the domain label $\hat{d}_i^o$. Again, we use the cross-entropy function as the loss function:

$$\hat{d}_i^o = \mathrm{Softmax}\left(\boldsymbol{W}_m \,\mathrm{ReLU}(\mathbf{Z}_i; \Theta_{dd}) + b_m\right), \tag{17}$$

$$L_o = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{D} \mathbf{d}_{ij} \log\left(\hat{\mathbf{d}}_{ij}^o\right) \tag{18}$$

Finally, the overall loss function for the domain discriminator is as follows:

$$L_{DD} = \alpha L_d + \beta L_o, \tag{19}$$

where α and β are the weight values.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3390748

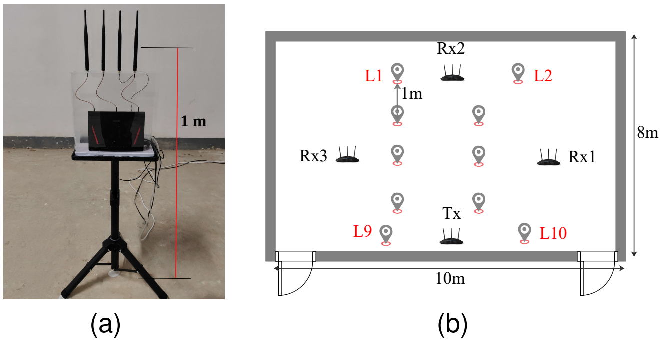


Fig. 6. We deploy one Tx and three Rx in a room of $8m \times 10m$, and select ten locations $L1 - L10$, the closest distance between them is less than $1m$.

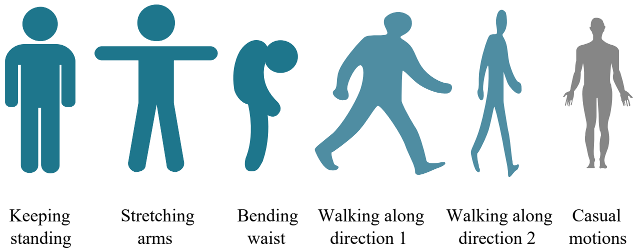


Fig. 7. We define five common motions, as well as casual motions, and collect the above six sets of CSI data for each person at each location.

E. Model Training

In our model, we need to minimize the loss L_{LP} of the location predictor to maximize the location prediction accuracy, while maximizing the loss L_{DD} of the domain discriminator so that domain-independent features are learned. Therefore, we get the overall loss function as shown below:

$$L_{all} = L_{LP} - L_{DD} = L_a + \gamma L_u - \alpha L_d - \beta L_o. \qquad (20)$$

We train our model with the augmented labeled data and unlabeled data, and iteratively update the model parameters during the training process. It is worth noting that in order to achieve the goals of our model, the loss function of the location predictor is taken positive during backpropagation, while the loss function of the domain discriminator is taken negative during backpropagation. The specific process is given in Algorithm 1.

V. Implementation

A. Experimental Setup

Hardware. We mainly build the hardware platform based on four ASUS RT-AC86U routers equipped with bcm4366c0 *network interface card* (NIC), one of which is used as Tx and three are used as Rx. The four routers have the same setup, *i.e.*, four transmit/receive antennas, and fixed on a tripod one meter above the ground, in order to better detect motions, as shown in Fig. 6a. During the data collection, we use an *industrial personal computer* (IPC) with Ubuntu 20.04 LTS operating system to remotely control four routers via Ethernet. In addition, we perform model training and prediction on a computer with Windows 10 system (Intel-i5 2.7GHz CPU, 32GB RAM).

Environment settings. As shown in Fig. 6b, the experimental environment is an empty room of $8m \times 10m$, and we place four routers near the four sides of the room. All tests cover ten different locations, marked $L_1, L_2, \cdots, L_{10}$, with the distance between the nearest two marked points being about 1m. It is worth noting that the environment remains unchanged during all tests, because changes in the environment also cause changes in the CSI fingerprints, but this paper does not analyze this.

People and motions. To collect CSI for different motions, we define six types of motions (*i.e.*, M1-M6) as shown in Fig. 7. These include five common motions, *i.e.*, five common motions that require the people to remain still during data collection. We also specifically declare a casual motion, *i.e.*, the people can do any motion in the marked location area during the data collection, such as turning in circles, squatting, shaking and etc. It is worth noting that we have two main reasons for defining these six detailed motions. Firstly, body movement changes the propagation path of WiFi signals on a large scale. The purpose of this paper is to eliminate this impact on fingerprint localization, and these six motions are easy to occur in people's daily lives. Secondly, it is clear that different motions define different domains, which facilitates verification of the effectiveness and robustness of *freeLoc*. In addition, the same motion performed by different people also have different effects. Therefore, we recruited six volunteers (*i.e.*, P1-P6) with different heights and weights to record the CSI of different people. Their height ranges from 160cm to 188cm, and their weight ranges from 55kg to 96kg.

B. Data Collection and Prepossessing

We install Nexmon CSI extractor on each of the four routers and collect CSI using injection/monitor mode [26]. The WiFi channel is set to 5GHz/80MHz, minus the subcarriers used for protection and empty load, the number of available subcarriers is $N_{sc} = 234$. Both the spatial stream and the core stream are four, and the number of Rx is 3, so $N_{ss} = 4 \times 4 \times 3 = 48$. We require each volunteer to do each motion for one minute at each location. We specify Tx to send 100 packets per second, and three Rx to monitor all packets.

We select 192 subcarriers for the convenience of data construction, and use 10 consecutive packets in the time series to obtain dynamic features, so the size of each data is $48 \times 192 \times 10 = 96 \times 96 \times 10$. We split the data into $6 \times 6 = 36$ domains according to people and motions, and randomly split the collected data for each domain into 50% training data and 50% testing data. In addition, for the training data, the source domains data is with location labels, while the target domains data is without location labels. Then we extract the CSI magnitudes of all available subcarriers for CSI fingerprints. In addition, all labels are encoded using one-hot.

C. Baseline Methods

In this paper, we compare the performance of *freeLoc* with the following baseline methods, and analyze the necessity of two key modules of *freeLoc*.

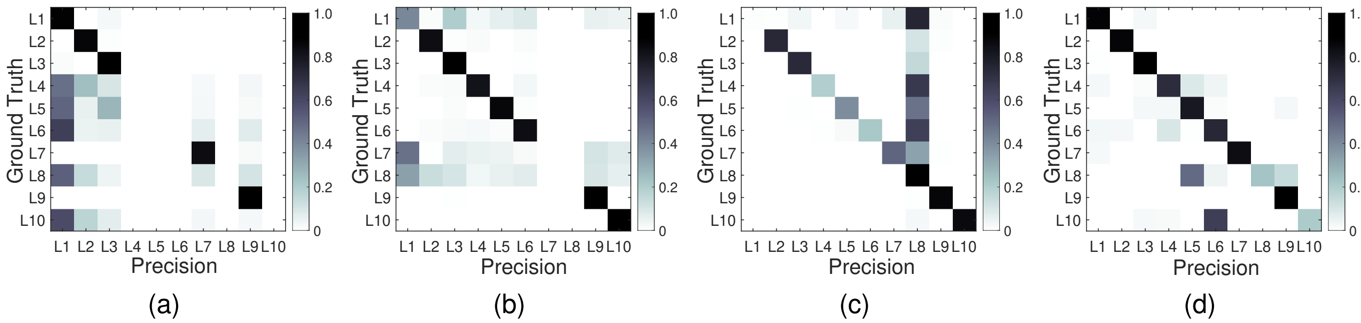


Fig. 8. The location prediction confusion matrices obtained after the predicted data of ten locations pass through (a) basic CNN, (b) AAE-only, (c) domain adaptation-only and (d) *freeLoc* respectively.

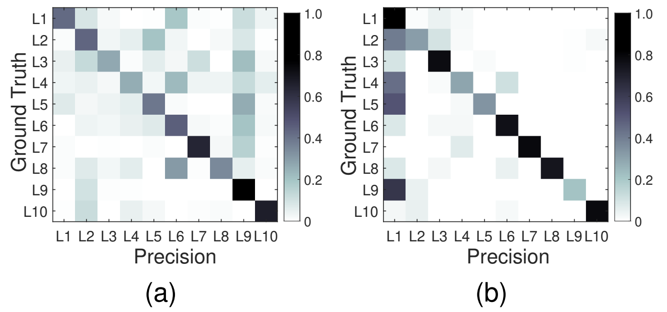


Fig. 9. The location prediction confusion matrices obtained after the predicted data of ten locations pass through (a) DAFI, (b) FiDo.

TABLE I
RECALL, PRECISION AND F1-SCORES FOR ALL LOCATIONS, USERS AND MOTIONS.

Method	Recall	Precision	F1-scores
Basic CNN	29.42%	49.06%	36.78%
AAE-only	58.44%	72.05%	64.63%
Domain adaptation-only	89.04%	61.91%	73.03%
DAFI	52.78%	57.15%	52.40%
FiDo	65.74%	81.00%	67.03%
freeLoc	**86.16%**	**84.32%**	**85.23%**

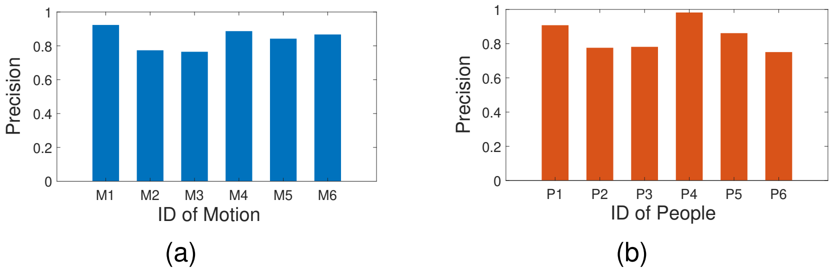
- **Basic CNN** is used as the baseline in this paper, that is, without data augmentation and domain adaptation, the labeled and unlabeled data are directly input into the CNN-based feature extractor to directly predict the location label.
- **AAE-only** implements only the AAEs of *freeLoc*, and directly connects them to the feature extractor and the location predictor without domain adaptation.
- **Domain adaptation-only** implements only the domain adaptation network of *freeLoc*, *i.e.*, directly feeds labeled and unlabeled data into the feature extractor, location predictor, and domain discriminator without AAEs.
- FiDo [21] takes into account the influence of different users, introduces data diversity using VAE, and learns domain-adaptive localization classifiers based on a special neural network with a joint classification reconstruction structure.
- DAFI [22] considers the impact of environment changes, uses two domain discriminators and pseudo-location labels for adversarial learning, and proposes a semi-supervised machine learning pipeline to solve the fingerprints inconsistency problem.

VI. EVALUATION

A. Localization Accuracy

In order to evaluate the performance of *freeLoc* as a whole, we first select three motions (M1, M4 and M6) of three volunteers (P1, P4 and P5) in the training data as source domains data, *i.e.*, with location labels, and the remaining volunteers or motions are used as target domains data, *i.e.*, without location labels, and all data have domain labels. Among them, domain adaptation-only and *freeLoc* are all able to use location-labeled data and location-unlabeled data to train their respective models, but basic CNN and AAE-only just use location-labeled data to train its model. Then we compare *freeLoc* with the above baseline methods. Table I presents the recall, precision and F1-scores for all locations, users and motions. It can be seen that the accuracy of basic CNN is extremely low because it only uses less labeled data for training. AAE-only and domain adaptation-only improve some accuracy by data augmentation and domain adaptation respectively, but there is still much room for improvement. In addition, since it only utilizes domain adaptation technology, DAFI is slightly less effective than FiDo in dealing with our problem, and both are weaker than *freeLoc*. Compared with these techniques, *freeLoc* can improve the precision by at worst 35%.

We also give the location confusion matrix of each method to better compare *freeLoc* with other methods, and the results are shown in Fig. 8 and Fig.9. From the results, compared to other schemes that cannot accurately distinguish multiple locations, *freeLoc* can have high classification accuracy in almost every location. Although the classification accuracy of a few locations is lower (*e.g.*, L8, L10), unlike other solutions, *freeLoc* does not have the situation where it is almost completely unable to distinguish different locations. In addition, it can also be seen from Table I that *freeLoc* has higher recall, precision and F1-scores at the same time.



(a) (b)

Fig. 10. Robustness of *freeLoc* to (a) six different motions (M1-M6) and (b) six different people (P1-P6).

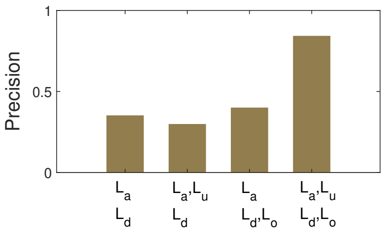
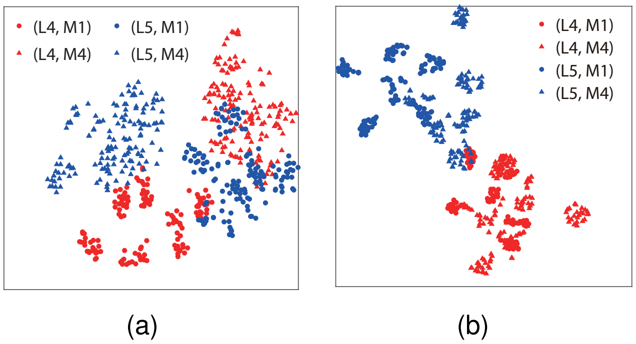


Fig. 11. Accuracy with or without the two constraints (L_u, L_o) of *freeLoc*.



(a) (b)

Fig. 12. Visualization results of location features after going through (a) basic CNN and (b) *freeLoc* respectively.

B. *Visualization of Location Features*

The goal of the deep learning model we propose in this paper is to learn a representation of location features independent of the users' motions. To evaluate the learned representations, we conduct the following experiments on the collected CSI dataset. Specifically, from the location-unlabeled data in the target domains, we select the data of two motions of two locations, *i.e.*, four domain-location pairs. We then randomly select 120 samples for each domain-location pair, and finally plot the learned representations of these samples by only the feature extractor and *freeLoc* according to Equ. 7, respectively.

We demonstrate on 2D space with t-SNE [31] and the results are shown in Fig. 12. It is worth noting that we use different colors (orange and blue) to represent different locations, and different shapes (circle and triangle) to represent different motions. As can be seen in Fig. 12, when basic CNN is used to extract location features, the features of the two locations are overlap due to the influence of different motions. In addition, the location features extracted by *freeLoc* are concentrated in two clusters, and there is almost no overlap between different locations. This demonstrates the effectiveness of the proposed deep model, *i.e.*, learning location features independent of the users' motions.

C. *Robustness to Different Motions*

In order to analyze the robustness of *freeLoc* in detail, we first divide the data into 6 groups according to different motions, and predict all the locations respectively (the source domains data comes from M1, M4 and M6 of P1, P4 and P5). Accuracy is shown in Fig. 10a. The results show that for the three kinds of motions including the source domains and the target domains, the location predictions are above 84%, while for the three motions that are all target domains, the prediction are still above 76%, although it has declined. Therefore, *freeLoc* maintains robustness to different motions.

D. *Robustness to Different People*

Furthermore, we divide the data into 6 groups according to different volunteers, and predict all the locations respectively (the source domains data comes from M1, M4 and M6 of P1, P4 and P5). Accuracy is shown in Fig. 10b. The results show that for the three users including the source domains and the target domains, the location predictions are above 86%, while for the three users that are all target domains, the prediction are still above 75%. Therefore, *freeLoc* maintains robustness to different people.

E. *Effect of Constraints*

Finally, we verify the impact of the two constraints we propose on the loss function in the model design, *i.e.*, L_u and L_o. Specifically, we perform model training for the following four situations in the loss function: (L_a, L_d), (L_a, L_u, L_d), (L_a, L_d, L_o) and (L_a, L_u, L_d, L_o), and then use the trained model to predict all locations, users, and motions. As shown in Fig. 11, the location prediction accuracy without two constraints is much lower than with two constraints. This shows that it is not feasible to directly use the original DANN network [23] to train our data, thus *freeLoc* is necessary.

VII. Related Work

In this section, we briefly review related work with *freeLoc* from the following two broad categories:

WiFi-based indoor localization. For more than a decade, device-oriented and device-free solutions have been developed for WiFi-based high-precision indoor localization and tracking, using information including RSSI and CSI. Common device-oriented localization ideas include AoA-based [8], ToF-based [16], fusion of AoA and ToF [9], RSSI fingerprints [17]. These device-oriented methods require users to carry smartphones, smart watches or other WiFi devices, and require dense deployment of WiFi APs in the environment, which is not convenient and practical in some scenarios. On the other hand, device-free localization schemes use WiFi signals reflected by the human body to estimate the user's location.

Their common ideas includes two aspects: measuring the parameters of the reflection path of the human body (AoA, AoD, Doppler and etc.) [32]–[34], and associating the characteristics of RSSI or CSI with the locations of the human body to create fingerprints [18], [19]. Among them, fingerprint-based methods generally need to collect fingerprints offline, and then locate users by matching real-time WiFi features with the collected fingerprints. In order to improve the localization accuracy and reduce the fingerprints collection cost, many deep neural network-based feature extraction, data augmentation, fingerprints library mapping and updating schemes have been studied [20], [21], [35].

Domain adaptation for WiFi inconsistency. RSSI and CSI are extremely sensitive to the surrounding environment. Changes in the environment, people of different sizes, different motions and even different transmitting/receiving devices may cause changes in reflected information, which is WiFi signal inconsistency [20]. Updating fingerprints in time can ensure the localization accuracy well, but collecting enough data is very expensive and impractical. In this case, it is necessary to use limited data to learn a transferable model. Therefore, domain adaptive learning is gradually being used in multiple perception systems including WiFi fingerprint localization to improve the learning performance in data-poor target domains by minimizing the data distribution difference between source and target domains. RSSI-based fingerprints localization systems show that the accuracy can be improved by utilizing a domain-adaptive-based approach to fingerprints collection time and device [36]–[38]. In CSI-based fingerprints localization, some works also use domain adaptive methods to combat the problem of CSI inconsistency caused by environmental changes and different users to improve accuracy [21], [22]. In addition, domain adaptation is also widely used in human activity recognition [30], object recognition [39] and etc.

Data augmentation for WiFi sensing. In WiFi sensing, data collection and labeling are very difficult. Because it cannot be easily observed like video data, WiFi signals cannot be directly understood. Therefore, WiFi sensing usually requires tedious data collection and labeling, which makes many solutions based on deep learning networks difficult to expand. Recently, some work has applied data augmentation schemes to WiFi sensing technology, aiming to synthesize a large amount of similar data through network models or physical mapping from a small labeling data. There are two typical WiFi sensing data generation methods. The first is to migrate data augmentation schemes from other fields to the WiFi sensing field [21], [40], [41], such as flipping and translation, VAE, and GAN. These methods directly convert labeled WiFi sensing data into the network to generate similar data that conforms to a specific distribution. The second method is to embed the theoretical model into the network learning model [42], [43], so that the generated data is more consistent with the actual WiFi channel. Although these data augmentation schemes show excellent performance in specific sensing tasks, they do not consider complex scenarios with highly coupled human motions.

VIII. Conclusion

In this paper, we propose *freeLoc*, a WiFi-based device-free fingerprints localization system, that aims to solve the problem of CSI fingerprints inconsistency caused by different users' motions. Specifically, *freeLoc* applies domain-adaptive deep learning techniques to our specific problem, regards known users' motions as the source domains, and takes the changed unknown users' motions as the target domains. To better learn location fingerprints features independent of users' motions, we make full use of the distribution of large unlabeled data, as well as the matching condition of location predictor and domain discriminator. In addition, to reduce the labeling cost of source domains data while ensuring the performance, we use data augmentation to obtain rich and diverse labeled data. Our experiments on real-world scenario show that *freeLoc* not only at worst improves accuracy 35% over other techniques, but also remains robust against multiple unlabeled different motions and users. In this way, *freeLoc* achieves the goal of reducing constraints on users' motions while maintaining accurate localization.

References

[1] Q. Huang and Y. Liu, “On geo-social network services,” in *2009 17th International Conference on Geoinformatics*. Ieee, 2009, pp. 1–6.

[2] T. Horozov, N. Narasimhan, and V. Vasudevan, “Using location for personalized poi recommendations in mobile environments,” in *International Symposium on Applications and the Internet (SAINT'06)*. IEEE, 2006, pp. 6–pp.

[3] H. Clougherty, A. Brown, M. Stonerock, M. Trepte, M. Whitesell, and R. Bailey, “Home automation and personalization through individual location determination,” in *2017 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, 2017.

[4] N. Wingfield, “Amazon moves to cut checkout line, promoting a grab-and-go experience,” *The New York Times*, vol. 5, p. 2016, 2016.

[5] R. T. Azuma, “A survey of augmented reality,” *Presence: teleoperators & virtual environments*, vol. 6, no. 4, pp. 355–385, 1997.

[6] M. Youssef and A. Agrawala, “The horus wlan location determination system,” in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, 2005, pp. 205–218.

[7] S. Kumar, S. Gil, D. Katabi, and D. Rus, “Accurate indoor localization with zero start-up cost,” in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014, pp. 483–494.

[8] J. Xiong and K. Jamieson, “{ArrayTrack}: A {Fine-Grained} indoor location system,” in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 71–84.

[9] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, “Spotfi: Decimeter level localization using wifi,” in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 269–282.

[10] D. Vasisht, S. Kumar, and D. Katabi, “{Decimeter-Level} localization with a single {WiFi} access point,” in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, 2016, pp. 165–178.

[11] J. Wang and D. Katabi, “Dude, where's my card? rfid positioning that works with multipath and non-line of sight,” in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, 2013, pp. 51–62.

[12] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, “Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices,” in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014.

[13] W. Wang, L. Mottola, Y. He, J. Li, Y. Sun, S. Li, H. Jing, and Y. Wang, “Micnest: Long-range instant acoustic localization of drones in precise landing,” in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022.

[14] J. Xu, H. Cao, D. Li, K. Huang, C. Qian, L. Shangguan, and Z. Yang, “Edge assisted mobile semantic visual slam,” in *IEEE INFOCOM 2020-IEEE Conference on computer communications*. IEEE, 2020, pp. 1828–1837.

[15] J. Xu, H. Cao, Z. Yang, L. Shangguan, J. Zhang, X. He, and Y. Liu, “{SwarmMap}: Scaling up real-time collaborative visual {SLAM} at the edge,” in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022.

[16] D. Vasisht, S. Kumar, and D. Katabi, “Sub-nanosecond time of flight on commercial wi-fi cards,” *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, p. 121–122, aug 2015. [Online]. Available: https://doi.org/10.1145/2829988.2790043

[17] C. Wu, J. Xu, Z. Yang, N. D. Lane, and Z. Yin, “Gain without pain: Accurate wifi-based localization using fingerprint spatial gradient,” *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 1, no. 2, pp. 1–19, 2017.

[18] 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*, 2012, pp. 183–196.

[19] J. Wang, H. Jiang, J. Xiong, K. Jamieson, X. Chen, D. Fang, and B. Xie, “Lifs: Low human-effort, device-free localization with fine-grained subcarrier information,” in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016, pp. 243–256.

[20] X. Chen, C. Ma, M. Allegue, and X. Liu, “Taming the inconsistency of wi-fi fingerprints for device-free passive indoor localization,” in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[21] X. Chen, H. Li, C. Zhou, X. Liu, D. Wu, and G. Dudek, “Fido: Ubiquitous fine-grained wifi-based localization for unlabelled users via domain adaptation,” in *Proceedings of The Web Conference 2020*, 2020, pp. 23–33.

[22] H. Li, X. Chen, J. Wang, D. Wu, and X. Liu, “Dafi: Wifi-based device-free indoor localization via domain adaptation,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 4, pp. 1–21, 2021.

[23] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[24] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial autoencoders,” *arXiv preprint arXiv:1511.05644*, 2015.

[25] D. Yan, Y. Yan, P. Yang, W.-Z. Song, X.-Y. Li, and P. Liu, “Real-time identification of rogue wifi connections in the wild,” *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6042–6058, 2022.

[26] M. Schulz, D. Wegemer, and M. Hollick, “Nexmon: Build your own wi-fi testbeds with low-level mac and phy-access using firmware patches on off-the-shelf mobile devices,” in *Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization*, 2017, pp. 59–66.

[27] Y. Grandvalet and Y. Bengio, “Semi-supervised learning by entropy minimization,” *Advances in neural information processing systems*, vol. 17, 2004.

[28] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, 2009.

[29] M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Conditional adversarial domain adaptation,” *Advances in neural information processing systems*, vol. 31, 2018.

[30] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas *et al.*, “Towards environment independent device free human activity recognition,” in *Proceedings of the 24th annual international conference on mobile computing and networking*, 2018, pp. 289–304.

[31] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.” *Journal of machine learning research*, vol. 9, no. 11, 2008.

[32] X. Li, S. Li, D. Zhang, J. Xiong, Y. Wang, and H. Mei, “Dynamic-music: Accurate device-free indoor localization,” in *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, 2016, pp. 196–207.

[33] Y. Xie, J. Xiong, M. Li, and K. Jamieson, “md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking,” in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[34] 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, pp. 1–22, 2017.

[35] X. Rao and Z. Li, “Msdfl: a robust minimal hardware low-cost device-free wlan localization system,” *Neural Computing and Applications*, vol. 31, pp. 9261–9278, 2019.

[36] Z. Sun, Y. Chen, J. Qi, and J. Liu, “Adaptive localization through transfer learning in indoor wi-fi environment,” in *2008 Seventh International Conference on Machine Learning and Applications*. IEEE, 2008, pp. 331–336.

[37] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, “Transfer learning for wifi-based indoor localization,” in *Association for the advancement of artificial intelligence (AAAI) workshop*, vol. 6. Citeseer, 2008.

[38] D. Li, J. Xu, Z. Yang, Y. Lu, Q. Zhang, and X. Zhang, “Train once, locate anytime for anyone: Adversarial learning based wireless localization,” in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[39] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, “Deep reconstruction-classification networks for unsupervised domain adaptation,” in *Computer Vision–ECCV 2016: 14th European Conference*. Springer, 2016, pp. 597–613.

[40] J. Zhang, F. Wu, B. Wei, Q. Zhang, H. Huang, S. W. Shah, and J. Cheng, “Data augmentation and dense-lstm for human activity recognition using wifi signal,” *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4628–4641, 2020.

[41] Q. Li, H. Qu, Z. Liu, N. Zhou, W. Sun, S. Sigg, and J. Li, “Af-dcgan: Amplitude feature deep convolutional gan for fingerprint construction in indoor localization systems,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 3, pp. 468–480, 2019.

[42] Z. Yang, Y. Zhang, K. Qian, and C. Wu, “{SLNet}: A spectrogram learning neural network for deep wireless sensing,” in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 1221–1236.

[43] X. Zhao, Z. An, Q. Pan, and L. Yang, “Nerf \textbf {2} : $Neuralradio-frequencyradiancefields,''$ *arXiv preprint arXiv:2305.06118*, 2023.