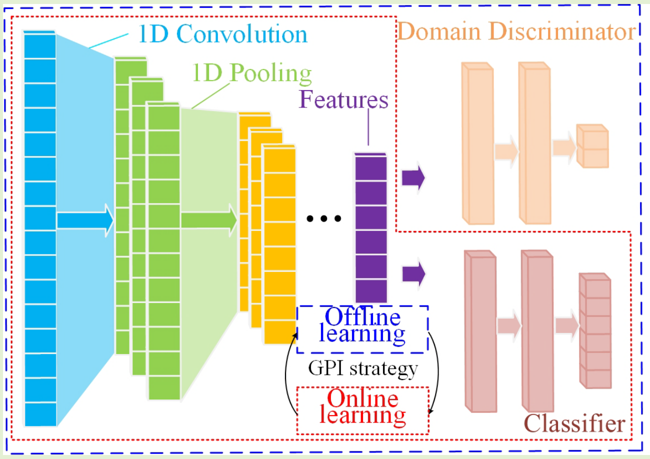
The Adaptive Fingerprint Localization in Dynamic Environment

Keliu Long, Chongwei Zheng, Kun Zhang, *Member, IEEE*, Chuan Tian, and Chong Shen

***Abstract*—Indoor localization service is an indispensable part of modern intelligent life, among which Wi-Fi based fingerprint localization system is popular in indoor positioning researches due to its advantages of low cost and widely deployment. However, Wi-Fi based localization system is susceptible to dynamic environment, and fingerprint collection and updating are time-consuming and labor-intensive. To address this problem, we propose a novel positioning framework based on multiple transfer learning fusion using Generalized Policy Iteration (GPI). Firstly, a 1-Dimension Convolutional Autoencoder (1-D CAE) is designed to extract features from one-dimensional fingerprint data; similar to Convolutional Neural Network (CNN), it can not only pay more attention to the information of different dimensions of fingerprints, but also compress redundant information and reduce noise. After that, Domain Adversarial Neural Network (DANN) and Passive Aggressive (PA) algorithm are fused to train localization model based on unlabeled fingerprint of target domain using the theory of GPI in offline stage. Finally, the model is fine-tuned with unlabeled fingerprints and few labeled fingerprints in daily online predictions to improve the performance of the localization system. Various evaluations in five typical scenarios validate the effectiveness of proposed algorithm in dynamic environment, with low tendency, easy recalibration, long-term stabilization high accuracy and so on.**



***Index Terms*—Feature extracting, fingerprint localization, generalized policy iteration, indoor localization, transfer learning.**

I. Introduction

LOCATION-BASED services (LBS) have gained great attentions in Internet of Things (IoT) and is becoming a research hotspot due to their basic requirements of intelligent applications, high-precision, and stable LBS for indoor applications [1]. In the outdoor positioning scenarios, the Global Navigation Satellite Systems (GNSS) can provide high-accuracy and low-latency localization services, which is common in our daily life based on the Global Positioning System (GPS). However, it is hard to use GNSS to get a satisfactory positioning result in a confined space due to the obstruction of line-of-sight that buildings cause on electromagnetic signal propagation [2]. In order to obtain indoor high-quality LBS, various technologies have been explored and applied in various indoor scenes, such as Ultra-Wideband (UWB) [3], Inertial Measurement Unit (IMU) [4], geomagnetism [5], Bluetooth [6], [7], Wi-Fi [8]–[10] and so on [11].

Theoretically, the UWB-based localization system has high positioning accuracy because of low-latency and anti-interference UWB signal, which estimates the target location using trilateral calculation. However, in practical application, the UWB localization system will be subject to Non-Line-of-Sight (NLoS) interference in harsh environment, resulting in large positioning error [12]. Hence, UWB anchors need to be added to meet the needs of accuracy positioning in

Manuscript received April 18, 2022; revised May 13, 2022; accepted May 13, 2022. Date of publication May 17, 2022; date of current version July 1, 2022. This work was supported in part by the High-Level Talent Project of Hainan Natural Science Foundation under Grant 2019RC236, in part by the National Natural Science Foundation of China under Grant 61861015, and in part by the Hainan Province Science and Technology Special Fund under Grant ZDKJ2021042 and Grant ZDKJ2021023. The associate editor coordinating the review of this article and approving it for publication was Prof. Danilo Demarchi. *(Corresponding author: Chong Shen.)*

Keliu Long and Chong Shen are with the State Key Laboratory of Marine Resource Utilization in South China Sea, and the School of Information and Communication Engineering, Hainan University, Haikou 570228, China (e-mail: keliulong@hainanu.edu.cn; chongshen@hainu.edu.cn).

Chongwei Zheng is with the Dalian Naval Academy, Dalian 116018, China (e-mail: chinaoceanzcw@sina.cn).

Kun Zhang is with the Education Center of MTA, Hainan Tropical Ocean University, Sanya 572022, China (e-mail: kunzhang@hntou.edu.cn).

Chuan Tian is with the Institute of Deep-Sea Science and Engineering, Chinese Academy of Sciences, Sanya 572000, China (e-mail: tianc@idsse.ac.cn).

Digital Object Identifier 10.1109/JSEN.2022.3175742

NLoS environment. Unfortunately, the cost of UWB system is considerably expensive, which limits application of UWB system in large-scale fields. In recent years. Micro-Electro-Mechanical System (MEMS) has made great progress, thus, the Inertial Measurement Unit (IMU)-based Inertial Navigation System (INS) is widely used in localization and navigation with low cost and power consumption [13]. The IMU is a self-constraint system, thus, it can estimate location through embedded accelerometer and gyroscope without any external information. However, due to the limited accuracy of commercial IMU and the characteristic of self-constraint system, the INS can only provide reliable localization results in a short time [14]. In pedestrian positioning scene, the IMU-based Pedestrian Dead Reckoning (PDR) algorithm is also adopted to locate people, however, the inaccurate estimations of step length and heading angle limit the accuracy of PDR algorithm [15]. As for geomagnetism, it will be strongly disturbed in the indoor environment with metal instruments, consequently, the geomagnetism cannot provide stable location-specific information. Compared with the formerly mentioned technologies, the Wi-Fi technology has its unique advantages.

Wi-Fi technology is widely researched in localization fields, it can be used for positioning in two ways: model-based ranging [16] and fingerprint matching [17]. Specifically, the model-based ranging method applies the path loss function to calculate distance from different bases (at least 3 bases in 2D plane or 4 bases in 3D space), and then drawing circles with radius with the calculated distance as the radius to get the intersection as the estimated location. Different from the model-based ranging method, the fingerprint matching method is divided into two stages: offline stage and online stage. In the offline stage, the signal features (fingerprints) and corresponding locations (labels) are collected in pre-positioning space and stored in a database. In the online stage, the newly collected fingerprints are compared with fingerprints stored in database to estimate location using various algorithms. Intuitively, the model-based ranging way is easier to implement than the laborious fingerprint matching method. However, in practice, compared with the fingerprint matching way, the model-based ranging way has larger localization errors and more drastic fluctuations due to its poor adaptability in the complex NLoS environment.

The fingerprint matching method not as negatively impacted by NLoS as the accuracy of fingerprint-based localization system mainly depends on the granularity of reference points in the database, the fingerprint matching technology, and the state of environment. As for the construction of reference points, it is a time-consuming and labor-extensive task. Many fingerprint matching technologies that have been designed to achieve indoor localization, such as K-nearest neighbors (KNN), weighted KNN (WKNN) [18], centroid algorithm, deep learning [19] etc. Among these technologies, Convolutional Neural Network (CNN) and CNN-based hybrid network based are widely used in various fingerprint matching tasks [20], because CNN can extract robust spatial features through convolutional manipulation in adjacent receptive fields. Besides, the fingerprint signal is location specific and the subcarriers of fingerprint is correlated [21], which is suitable for processing with CNN. Although CNN can theoretically provide high-accuracy fingerprint matching results, its performance will be greatly limited in real-world dynamic environments. These such algorithms, however, are typically degraded in dynamic environments which influences the localization, such as furniture layout, human movement, states of door or window (opening or closing) [22] and so on. In a word, it is a challenge to maintain stable and reliable location results in dynamic environment with Wi-Fi localization systems. Therefore, we can improve the accuracy of indoor positioning results through optimizing fingerprint database, matching technology, and environment modeling. In fact, the above three optimization schemes can be resolved by refreshing labeled fingerprints in database. Unfortunately, the collection of labeled fingerprints is also a time-consuming and laborious work, which limits the application range of fingerprint-based positioning system.

As stated above, the cost of refreshing labeled Wi-Fi signal in a database is an obstacle for Wi-Fi based matching localization system in dynamic scenes. In order to eliminate this situation, various techniques are proposed to improve stabilization of fingerprint positioning system. Domain Adaptation (DA) in Transfer Learning (TL) is popular in solving fingerprint adaptation in dynamic environment [23]. The approach transfers the features of source domain (the initial environment) and target domain (the changed environment) into the same feature space using supervised or semi-supervised learning. In doing so, the previously invalid labeled fingerprints can be used in the changed environment, which reduces the dependence on labeled data in the changed environment. Although the DA alleviates the reliance on labeled data, it still needs a certain number of labeled fingerprints to calibrate the positioning system, especially in large-scale field. The Domain-Adversarial Neural Network (DANN) [24] can solve this problem, because it's an unsupervised DA neural network, i.e., only unlabeled data is required in DA. However, the unsupervised learning in DA is not completely suitable for positioning scenarios, once the DA algorithm finishes training, the DA stops whereas the change of environment is inevitable. Nonetheless, most fingerprint DAs focuses on improving the performance of the algorithm rather than tracking the dynamic environment characteristics.

Most aforementioned DA researches are offline learning algorithms where data is processed in batch form; however, the fingerprint data is collected in chronological order and the environment changes all the time, which will result in inconsistency of data distributions in training data. Although the unsupervised DANN can alleviate inconsistent distribution of data by collecting large amounts of unlabeled data in the target domain, the DA between former distribution and current distribution cannot be solved once and for all due to unpredictable change of environment. The online DA algorithm can calibrate the positioning system with a small amount of sequential labeled data, which is consistent with the labeled data collection, i.e., sequential processing way. The Passive Aggressive (PA) method [25] is a kind of online transfer learning algorithm that processes the received data in sequential manner; it is proved to be an excellent lightweight online DA algorithm, where the hinge loss function is adopted as an indicator to realize a classification or regression mission. In addition, the PA algorithm can track the distribution change of fingerprints, if the environment changes, the PA algorithm will gradually forget the previous weights and learn new weights.

In this paper, in order to solve the problem of difficult acquisition of labeled data and inconsistent distribution of collected fingerprints, the DANN is adopted to realize DA in offline stage; in online stage, the PA algorithm tracks dynamic characteristics of the environment and calibrates the entire positioning system. The main contributions of this paper are listed as follows:

- DANN and PA algorithm are used separately or jointly in different stages of the positioning system. The DANN can realize initial DA with unlabeled data, where the unlabeled data is easier to obtain than labeled data and the 1D Convolutional Autoencoder (1D-CAE) is embedded into DANN to extract input features; the PA algorithm processes data in sequential manner for calibrating positioning system, which is consistent with the order of labeled data collection.
- A uniform positioning framework based on online and offline DA is constructed learning from the ideas of Global Policy Iteration (GPI) [26], this means that the DANN and PA algorithm are not simply cascaded together but trained in a uniform framework. The proposed framework is comprised of individual modules, which means that the online and offline modules in this framework can be replaced by other advanced algorithms in various mission sets.
- Various experiments are conducted to verify the validity of the proposed framework; the results show that the proposed method can give long-term stability and high-accuracy positioning services.

The remainder of the paper is organized as follows: Section 2 reviews the positioning and DA related works. Section 3 introduces online and offline learning used in this paper. In Section 4, the proposed framework and algorithm of adaptive positioning are described in detail. Section 5 presents the evaluation experiments of the adaptive localization method. Section 6 gives the conclusions of our work.

II. Related Work

A. Traditional Positioning Technologies

Zhao *et al.* [27] used crowdsourcing and multi-source fusion to reduce laborious radio map construction, where the PDR and EKF-based multi-source fusion algorithm are used to improve the accuracy and robustness of localization results. Li *et al.* [28] proposed unsupervised wireless positioning system based on Deep Reinforce Learning (DRL) frame and Received Signal Strengths (RSS), where the localization process is modeled as a Markov Decision Process (MDP), it has good performance in the 100m level spatial scale. Similarly, the DRL based on Wi-Fi signals is also adopted to bisect localization space and calculate location with flexible localization resolution in reference [29], it has reduced the searching complexity and realized pseudo 3D localization (multi-floor positioning). In reference [22], the spatial-temporal focusing effect of retransmitting time-reversal signals, namely time-reversal resonating strength (TRRS), is used to realize high-resolution positioning, moreover, the Autoencoder (AE) is trained to record current features of environment; after that, the trained AE calibrates the newly received signals, and the location is calculated by retransmitting signals. Although this works has considered the effects of dynamic environment on positioning accuracy, it will be limited by dramatically in a dynamically changing environment. The TRRS is also utilized in reference [30] under NLoS scene, where the frequency and spatial diversities are exploited for centimeter-accuracy positioning in relatively stable environment. Zhao *et al.* [31] used Convolutional Autoencoder (CAE) to extract features of raw received signal strength indication (RSSI), and then the classification neural network is cascaded with CAE to estimate the position. A hybrid fusion positioning scheme is proposed based on the complementarity between Wi-Fi and pedestrian dead reckoning (PDR) in reference [32], where the Machine Learning (ML) is utilized to remove the outliers of received signal strength (RSS); it also reveals that Wi-Fi signals have clustering characteristics, i.e., most of the received Wi-Fi strengths concentrate around the true value with a threshold.

In the traditional fingerprint localization system, the localization system is affected by the multipath effect which seriously reduce the stability of positioning system. Thus, Song *et al.* designed a positioning system based on dual-channel convolutional neural network, named DuLoc, to estimate location using CSI [20]. In reference [33], a CNN-LSTM hybrid model is proposed to provide stable localization results using both temporal information (sliding window processing) and spatial information (converting sequence to picture) of CSI signals, and the positioning accuracy is about 2.5 meters. Different from most works, the reference [34] concentrates on constructing robust positioning characteristics, where the phase differences and amplitude differences of CSI are used to construct three gray images, after that, the three grayscale images are fused into one RGB image for CNN identification and positioning. In reference [35], the depth-wise separable convolution is used to simplify CSI-based localization model, which can reduce latency and improve performance of the system. The CSI of Multiple-input multiple-output (MIMO) is also investigated in [36]–[38], the main contribution of these works is to construct suitable CNN-based neural structures and composite source fingerprints, which can extract deep features of CSI signals and provide high-precision positioning results in limited dynamic environment. Wang *et al.* innovatively proposed the use of angle of arrival (AoA) for indoor positioning, where the AoA is estimated by CSI signal of 5G Wi-Fi. Specifically, the images constructed with estimated AoA are used for Deep Convolutional Neural Network (DCNN) training in offline stage. In the online stage, the real-time CSI AoA images are used to predict all location probabilities. After that, the target location is estimated through weighted average of the R largest outputs [39].

The above-mentioned research works show that the traditional ML methods are widely used in positioning system with supervised learning, it can be treated as a static characteristic learning in which the parameters of network are fixed after finishing training; consequently, the traditional ML under supervised learning is not suitable for dynamic environment positioning. Therefore, various methods are proposed to improve the performance of positioning system in changing environment including sensor fusion and DA. There are already some positioning systems based on DA. The theory of domain adaption suggests that, for effective domain transfer goal, the features extracted from source domain and target domain should be hard to distinguish.

B. Domain Adaptation in Fingerprint Localization

Building a mapping between source domain and target domain is known as DA when there is a shift (change) in domain distribution. After that, the classifier or regression relationship learned for the source domain is also suitable for the target domain. The mapping in DA can be realized in the situation that target domain data with few labels or without any labels, so that the DA can be divided into two types: unsupervised DA and semi-supervised DA. Zhang *et al.* proposed an improved TrAdaBoost to realize CSI phase fingerprint adaptation given labelled source data and target data, and the One-vs-Rest algorithm and One-Hot coding were also used to enhance the robustness of positioning results [40]. Lin *et al.* designed an online transfer learning framework based on Long Short-Term networks (LSTM) feature extraction for time-varying distribution DA. The approach combined the proposed transfer learning model with the ensemble approach to avoid overfitting and unfitting problems. In reference [23], the DA in localization based one-dimension Convolutional Neural Network (1D-CNN) is realized with Semantic Alignment (SA), moreover, a domain selection model is also trained to change pattern of system, and thus the localization results are more reliable. In reference [41], features squeezing and Class Alignment (CA) Loss are utilized to maximize the distance between different classes in positioning system.

It can be seen that the above solutions mostly perform semi-supervised or supervised DA, while fewer methods perform completely unsupervised DA in Wi-Fi fingerprint localization systems, moreover, the online transfer learning and offline transfer learning are used separately in these works. Although it can achieve high-precision positioning in the early stage under the premise that the target domain has sufficient labelled data, it has two limitations: the offline adaptation results cannot work well in dynamic environment, and the labelled data are relatively few or none. Furthermore, the online DA will discard source domain information, which only depends on the current input data, while the offline DA ignores the change of the dynamic environment resulting in instability of localization system.

C. CSI and RSSI Fingerprints

The Wi-Fi based fingerprint positioning system are mainly divided into two schemes: RSSI based and CSI based. In the last decade, the RSSI-based indoor positioning system is very popular in industry or academia due to its lost cost, easy access. There are two ways to use RSSI for positioning: ranging calculation and fingerprint matching. In ranging calculation, Log-normal Distance Path Loss (LDPL) model is used to calculate distances from interesting place to Wi-Fi devices, after that, the estimation algorithm (e.g. trilateration algorithm) uses the ranging results to estimate the location. For example, Bo Yang *et al.* proposed a trilateration algorithm based on value theory for RSSI-based indoor localization [42]. It should be noted that at least 3 or 4 ranging results are required in 2D plane or 3D space. The disadvantages of ranging and positioning based on RSSI are that the ranging model is easy to be disturbed by dynamic environment (multipath effect), and the model parameters are difficult to obtain (massive data fitting). RSSI based fingerprint matching is consisted of offline fingerprint collection and online fingerprint matching; with the development of deep learning, fingerprint positioning technology based on RSSI has gradually become a research hotspot. For example, Li *et al.* proposed a RSSI-based localization system which has long-term stable performance, the key technique used is that three classification modules are utilized to vote for pseudo labels for system training based on DANN; at the same time, the authors also applied output smearing to maintain the diversity among three modules [43]. In reference [44], three CNN networks are used as encoders to preprocess RSSI fingerprint data, so that fingerprint features with the same label are closer, while fingerprint features with different labels are farther apart, which has high positioning accuracy and performs better than the general schemes in a long-time scale. Although these studies have shown that the positioning system based on RSSI fingerprints can achieve long-term high-precision localization, this is mainly due to the powerful feature extraction ability of the designed network and a certain number of Wi-Fi devices. Essentially, RSSI is a coarse-grained positioning signal which is the result of the superposition of a series of multipath signals and is vulnerable to multipath effect.

With the improvement and addition of the local area network (LAN) technical standards (EEE 802.11a/g/n protocol), CSI can be obtained from commercial Wi-Fi devices [45], where it has fine-grained descriptions of transmission channel. Therefore, in recent years, CSI has been widely used in the field of indoor positioning [46]. The essence of CSI is the accurate description of the channel, which can provide amplitude and phase of subcarrier. Therefore, from the perspective of information, more detailed description means more information, thus CSI has greater potential to obtain robust location-specific features [47]. Moreover, the RSSI-based localization system needs large number of Wi-Fi devices to extract robust location-specific information, which may increase positioning cost. At the same time, it hard to keep all device in valid reception range according to overlapping area of all Wi-Fi devices. Therefore, the CSI of Wi-Fi signal is utilized for positioning in this paper, and a more detailed comparison of CSI and RSSI can be found in reference [47].

In order to fully use the advantages of online and offline DA and explore the characteristics of fingerprint positioning, a joint positioning framework based on online and offline transfer is proposed to realize high-accuracy localization and

TABLE I
NOTATIONS USED IN THIS PAPER

Notations	Meanings
D_S	source domain
D_T	target domain
X_S	source domain feature space
Y_S	source domain label space
X_T	target domain feature space
Y_T	target domain label space
$P(X_S,Y_S)$	source domain feature distribution
$P(X_T,Y_T)$	target domain feature distribution
S	source sample data (labeled)
T	target sample data (unlabeled)
$\mathbf{x}_i$	the i-th feature instance
$\mathbf{y}_i$	the label of i-th feature instance
D_T^X	the marginal distribution of target domain D_T over X
R_{D_T}	the risk on D_T
d_H	H -divergence
$\hat{d}_A$	Proxy A -distance
γ	the error of optimal classifier
$C(\cdot)$	binary classifier
$h^{(j)}(\cdot)$	predictor of PA algorithm
$\mathbf{w}_j$	weight matrix on j-th class
K	the number of classifications
$l(\cdot)$	the loss function
$\Psi_{\{j=y_i\}}$	indicator function
SP	support vector
$T_f(\cdot;\theta_f)$	feature extractor with parameter θ_f
$T_d(\cdot;\theta_d)$	domain discriminator with parameter θ_d
$T_p(\cdot;\theta_p)$	predictor with parameter θ_p

long-term robustness. More specifically, the DANN is used to realize fingerprint DA with unlabeled target-domain data in the offline stage, it should be noted that the DANN can also process labelled data if available. In addition, the PA algorithm is adopted to calibrate fingerprint positioning system with few sequential labeled data in online stage. The DANN, PA algorithm and proposed positioning framework will be discussed in later sections.

III. PRELIMINARIES

A. CSI Fingerprint

Channel State Information (CSI) is the characteristic of transmission link, it gives fine-grained description about channel states including amplitude and phase information. The transmission model in frequency domain can be expressed as

$$\boldsymbol{Y} = H\boldsymbol{X} + N \tag{1}$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are respectively transmitted signal and received signal, N is corresponding Gaussian noise, H denotes CSI function which is a plural:

$$H = |H|\, e^{i\angle H}, \tag{2}$$

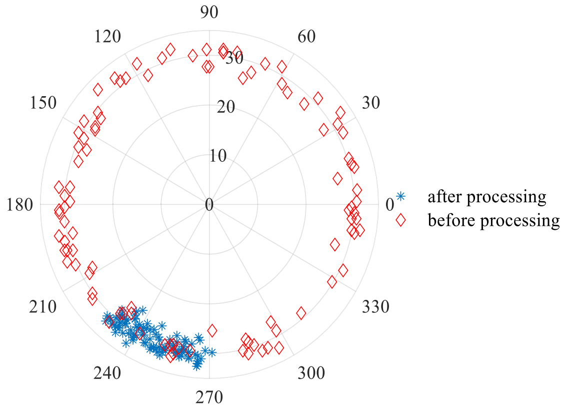


Fig. 1. The phases and amplitudes of same subcarrier in same place before and after processing.

where $|H|$ is the amplitude and $\angle H$ denotes the phase. However, due to noise interference and frequency bias and drift, the received raw phases are random. Therefore, the raw phase cannot be directly used to estimate location. According to reference [48], the relatively stable phase of each subcarrier after linear transformation can be represented by

$$\tilde{\phi}_i = \hat{\phi}_i - ki - b, \tag{3}$$

where $\tilde{\phi}_i$ and $\hat{\phi}_i$ are the processed phase and the raw received phase of i-th subcarrier, respectively. k and b can be expressed as

$$k = \frac{\hat{\phi}_L - \hat{\phi}_1}{L-1}, \tag{4}$$

$$b = \frac{1}{L}\sum_{i=1}^{L} \hat{\phi}_i, \tag{5}$$

where L is the total number of subcarriers. The Fig. 1 compares the phases and amplitudes of same subcarrier in same place before and after transformation, it shows that the processed phases are more distinguishable and stable compared with the raw phases, and thus the processed phases are more location-specific. The fluctuations in phase and amplitude are caused by the dynamic characteristics of the environment.

B. DA Problem Statement

The fingerprint-based localization system needs to refresh its fingerprint database once the environmental changes including scene changes, changes in environmental parameters (temperature, humidity, state of window or door), pedestrian interference, etc. As shown in Fig. 2, the CSI varies with time, and thus, the former CSI should be adapted into new domain to improve the stability and reliability of positioning system. The former CSI fingerprint domain is called source domain, and the current CSI fingerprint domain is the target domain.

Suppose D is the joint distribution domain over $X \times Y$, where X is the input feature space (fingerprints), Y denote the label space (location coordinates). Then a domain is defined as $D = \{X, Y, P(X, Y)\}$, where $P(\cdot)$ is the joint distribution function, $X \in X$, $Y \in Y$.

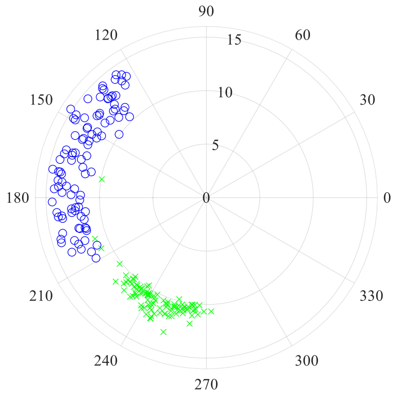


Fig. 2. The received CSI of same place in different time.

In unsupervised task, the labelled source data S and unlabeled target data T can be respectively represented as

$$S = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n} \sim (D_S)^n, \quad D_S = \{X_S, Y_S, P(X_S, Y_S)\}, \tag{6}$$

$$T = \{\mathbf{x}_i\}_{i=n+1}^{n+m} \sim (D_T^X)^m, \quad D_T = \{X_T, Y_T, P(X_T, Y_T)\}, \tag{7}$$

where the labelled source data $S = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ are sampled from source domain D_S, the target data are sampled from D_T^X (the marginal distribution of target domain D_T over X). The superscripts n and m are the sample number of source domain and target domain, respectively.

In the localization system, we have obtained the prediction function $f(\cdot)$ (classification or regression) with the source data S in the initial stage:

$$Y_S = f(X_{Si}). \tag{8}$$

The aim of DA is to improve the prediction ability of function $f(\cdot)$ using unlabeled data T in target domain D_T, thus the aim of DA is $\min_{(\mathbf{x},\mathbf{y})\sim D_T} [-\sum_{i=n+1}^{n+m} \mathbf{y}_i \log(f(\mathbf{x}_i))]$ in classification task or $\min_{(\mathbf{x},\mathbf{y})\sim D_T} (\sum_{i=n+1}^{n+m} |f(\mathbf{x}) - \mathbf{y}|^2)$ in regression task, while the labels of D_T are missing.

1) Offline Optimization: In DANN, the aim of DA is also can be represented as optimizing the function $f(\cdot)$ with low target risk [24]:

$$R_{D_T} = \Pr_{(\mathbf{x},\mathbf{y})\sim D_T} (f(\mathbf{x}) \neq \mathbf{y}), \tag{9}$$

$$R_{D_T} = |f(\mathbf{x}) - \mathbf{y}|/|\mathbf{y}|, \quad (\mathbf{x}, \mathbf{y}) \sim D_T. \tag{10}$$

And then,

$$R_{D_T}(C) \leq R_{D_s}(C) + d_H(D_S^X, D_T^X) + \gamma, \tag{11}$$

$$d_H(D_S^X, D_T^X) = 2 \sup_{C\in H} \left| \Pr_{\mathbf{x}\sim D_S^{\mathbf{X}}}[C(\mathbf{x}) = 1] - \Pr_{\mathbf{x}\sim D_T^{\mathbf{X}}}[C(\mathbf{x}) = 1] \right|, \tag{12}$$

where d_H is H-divergence to qualitatively describe the difference between source domain and target domain; the hypothesis class H is a set comprised of binary classifiers $C(\mathbf{x}) \to \{0, 1\}$, and γ is the error of the optimal classifier. In practical applications, the calculation condition of H-divergence is too harsh, and as such, it can instead be approximated by Proxy A-distance:

$$\hat{d}_A = 2(1 - 2\varepsilon), \tag{13}$$

where the generalization error ε is given by discriminating source data and target data. Therefore, in order get low risk R_{D_T} in the target domain (high-accuracy estimated positioning results), we need to reduce H-divergence while maintaining low source domain risk, i.e., increase the domain discrimination error (fooling the discriminator) through data feature mapping or extracting, and keeping high-accuracy predicted results in the source data.

From the previous summary, it can be seen that the Domain-Adversarial Neural Networks (DANN) is comprised of three parts:

- a feature representation part extracts features from both the source domain and the target domain,
- the label predictor (classification or regression) is used in all stages including training and testing, and lastly a
- domain classifier is used to judge whether the feature comes from the source domain or the target domain.

In the training stage, the label predictor is optimized through minimizing its loss on training samples, while the feature representation part is optimized for the purpose of maximizing the loss of domain classifier and minimizing the loss of label predictor which can extract the domain-invariant features.

Many DA methods match the corresponding labels of the source data and the target data by reducing the dimensionality feature expressions in the two domains (source and target). This means that the source domain has the same or more categories (labels) than the target domain. In other words, most works only consider that the difference on feature distribution $P(X)$ between source domain and target domain (i.e., $P(X^S) \neq P(X^T)$), rather than the difference of joint distribution of feature and label on $X \times Y$ (i.e., $P(X^S, Y^S) \neq P(X^T, Y^T)$). In fingerprint-based localization systems, these DA methods could only be used in same area (i.e., $P(X^S) \neq P(X^S)$, $D_T^Y \subseteq D_S^Y$). However, the DANN training structure provides another possibility, that is, the target data domain can have more categories (labels) than the source data domain (i.e., $P(X^S) \neq P(X^S)$, $D_T^Y \supseteq D_S^Y$), similar to the idea of label embedding. Therefore, the DANN-based DA can considerably reduce the dependence on the labelled data in target domain (the labelled data in target domain could be also used in DANN training when they are available).

2) Online Optimization: Although the offline transfer learning works well in fingerprint DA, the method is a time-consuming process and cannot capture the dynamic characteristics of the environment. Besides, offline transfer learning does not match the form of data collection, i.e., offline transfer learning is batch data used for training at the same time, while the data are chronologically collected. Therefore, it is necessary to embed the online transfer learning into a positioning algorithm framework for tracking the dynamic features of the environment. The online transfer learning is designed to deal with the situation that the target data is generated one by one (or in a sequential manner), because of inefficient fingerprint collecting (lasting several minutes on a collecting point). In practice, it's hard to get large amounts of labelled data in the target domain (waiting for fingerprint refreshing section or newly different space).

Since the Passive Aggressive (PA) algorithm processes the received data in a sequential manner, it has the ability of tracking the distribution change of data. If data is drawn from same distribution (stable environment), the PA algorithm will keep optimizing (data is disturbed by unavoidable noise) with minor modification, when data is drawn from different distributions, the PA algorithm will slowly forget former weights and gradually learn a new distribution.

In the online transfer learning, the estimation function is trained with single instance in a sequential manner, thus the estimate functions used in classification and regression missions can be respectively represented as

$$h^{(i)}(\mathbf{x}) = \arg\max_{j=1,\ldots K}(\mathbf{w}_j^{(i)} \cdot \mathbf{x}), \tag{14}$$

$$g^{(i)}(\mathbf{x}) = \mathbf{w}^{(i)} \cdot \mathbf{x}, \tag{15}$$

where the discriminative function $h^{(i)}(\cdot)$ predicts label of input $\mathbf{x}$ in the i-th round learning, $\mathbf{w}_j^{(i)}$ is the weight vector for calculating the probability (score) of class j; the regression function $g^{(i)}(\cdot)$ directly estimate output value. Suppose input instance $\mathbf{x} \subset \mathbb{R}^D$ is D-dimensional feature, the weight vector is $1 \times D$ dimension in classification task and $n \times D$ dimension in regression task, where n is related with a specific mission.

In a classification mission, the classification loss is represented with hinge loss function:

$$l(\mathbf{w}^{(i)}) = \max\left[0, 1 - \left(\mathbf{w}_{y_i}^{(i)} \cdot \mathbf{x}_i - \max_{j \neq y_i} \mathbf{w}_j^{(i)} \cdot \mathbf{x}_i\right)\right], \tag{16}$$

where the term $\mathbf{w}_{y_i}^{(i)} \cdot \mathbf{x}_i - \max_{j \neq y_i} \mathbf{w}_j^{(i)} \cdot \mathbf{x}_i$ denotes the minimum distance (margin) from right class y_i to any other class j. This shows that the weight will be not updated when the loss function is zero, i.e., $\mathbf{w}_{y_i}^{(i)} \cdot \mathbf{x}_i - \max_{j \neq y_i} \mathbf{w}_j^{(i)} \cdot \mathbf{x}_i \geq 1$. Hence the weight update is realized by solving an optimization function of the form:

$$w^{(i+1)} = \arg\min_{\mathbf{w}_j} \frac{1}{2} \sum_{j=1}^{K} \left\| \mathbf{w}_j - \mathbf{w}_j^{(i)} \right\|^2 \quad s.t.\ l(\mathbf{w}^{(i)}) = 0. \tag{17}$$

However, in a practical application, it's hard to exactly solve the optimize function due to the inevitable noise in training data. Hence the slack variable was added into the former constraints to get other two optimization functions: PA-I and PA-II [25].

In this paper, the PA-I is adopted to learn environment characteristics as follows:

$$\mathbf{w}^{(i+1)} = \arg\min_{\mathbf{w}_j} \frac{1}{2} \sum_{j=1}^{K} \left\| \mathbf{w}_j - \mathbf{w}_j^{(i)} \right\|^2 + C\xi, \\ s.t.\ l(\mathbf{w}^{(i)}) \geq 1 - \xi, \quad \xi \geq 0, \tag{18}$$

where positive C is the hyper-parameter, the larger the C, the more aggressive the PA-I algorithm. According to reference [49], the closed-form PA-I update using support classes for multiclass mission is

$$\mathbf{w}_j^{(i+1)} = \mathbf{w}_j^{(i)} - \tau_j \left(1 - \Psi_{\{j=y_i\}}\right) \mathbf{x}_i + \Psi_{\{j=y_i\}} \sum_{j \neq y_i} \tau_j \mathbf{x}_i, \tag{19}$$

$$\tau_j = \|\mathbf{x}_i\|^{-2} \max \\ \times \left[0, l_j - \max\left(\sum_{u \in S} \frac{l_u}{|S|} + \frac{C}{|S|} \|\mathbf{x}_i\|^2, \sum_{u \in S} \frac{l_u}{|S|+1}\right)\right], \tag{20}$$

where j is class label, $\Psi_{\{j=y_i\}}$ denotes indicator function, the term l_j is defined as $l_j = \max\left[0, 1 - \left(\mathbf{w}_{y_j}^{(i)} \cdot \mathbf{x}_i - \mathbf{w}_j^{(i)} \cdot \mathbf{x}_i\right)\right]$, SP denotes the support vectors which is determined by

$$\sum_{j=1}^{k-1} l_{\sigma(j)} < \min\left(\frac{(k+1)C}{\|\mathbf{x}_i\|^2} - l_{\sigma(k)}, k l_{\sigma(k)}\right), \tag{21}$$

where $\sigma(k)$ is the k-th class in the descending order of l_j, $j = 1, \ldots, K$.

In regression mission, the prediction loss is expressed as

$$l(\mathbf{w}^{(j)}) = \max\left(0, \left|y_j - g^{(j)}(\mathbf{x}_j)\right| - \varepsilon\right) \\ = \max\left(0, \left|y_j - \mathbf{w}^{(j)} \cdot \mathbf{x}_j\right| - \varepsilon\right). \tag{22}$$

Hence the corresponding weight update rule is

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} + \frac{\max\left(0, \left|\left|y_i - \mathbf{w}^{(i)} \cdot \mathbf{x}_i\right| - \varepsilon\right|\right)}{\|\mathbf{x}_i\|^2 + \frac{1}{2C}}, \tag{23}$$

As is stated above, the mechanisms (update rule, data form, etc.) between offline DANN and online PA are different from each other. Building a bridge between offline learning and online learning can achieve better fingerprint domain transfer and environment characteristic tracking. Besides, the combination of DANN and PA fully conforms to the characteristics of fingerprint positioning, i.e., the initial stage is unlabeled data and the positioning stage is few labelled data with sequential manner.

IV. Adaptive Wi-Fi Based Localization

A. 1D-CAE Feature Extractor

In the positioning system, the received raw location-specific fingerprints are multi-dimensional and noise-contaminated due to inherent characteristics of Wi-Fi signals, and the outliers will degrade the performance of positioning system. Therefore, the autoencoder is usually used to denoise and compress signals, because the autoencoder can extract the deep characteristics of input signals under unsupervised learning.

After finishing training, the learned features should be domain-invariant and prediction-discriminative. In addition, due to 1D feature of fingerprint, the experiments in reference [23] show that 1D-CNN outperforms 2D-CNN and DNN in Wi-Fi fingerprint signal processing. Therefore, in order to integrate merits of AE and 1D-CNN, the 1D-Covolutional Autoencoder (1D-CAE) shown in Fig. 3 is designed to refine the signal and get reliable and robust characteristics. Thus, it abstracts not only space information between subchannels of signal but also deeper feature representations, which will

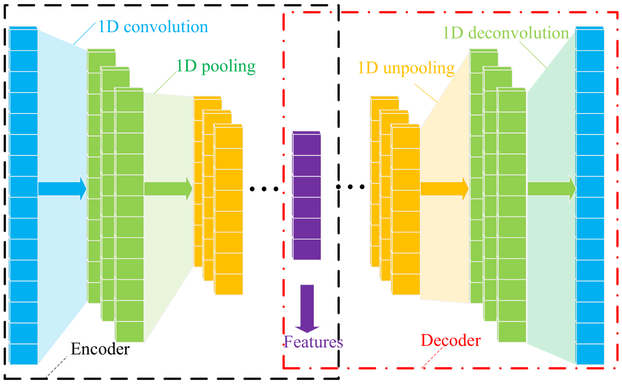


Fig. 3. The structure of 1D-CAE, which is suitable for processing 1D data. The multiple 1D intermediate layers (green and yellow) are results of different 1D convolutional kernels.

facilitate the extraction of domain-invariant and prediction-discriminative features. Furthermore, it can denoise the raw input signal after unsupervised learning. 1D-CAE has two parts including encoder and decoder. In encoding process (black dash line), 1D kernels are used to process 1D fingerprint data similar to 2D-CNN where the convolutional and pooling kernels are two-dimensional. In the unpooling manipulation of encoder part (red dash-dot line), the nearest interpolation method is adopted to restore the loss information of the previous max pooling stage. The deconvolution can be expressed as

$$I = f(\mathbf{k}_1{}^*\mathbf{O}_1 + \ldots + \mathbf{w}_{k_n}{}^*\mathbf{O}_{k_n}), \tag{24}$$

where I is the input of current layer, $\mathbf{k}_i$ and $\mathbf{O}_i$ are respectively the i-th convolutional kernel and the output of the last layer, the subscript k_n denotes the number of kernels.

B. The Adaptive Fingerprint Transfer

As is stated above, offline and online transfer learning have different working mechanism. More specifically, offline weight update needs batch data, while online weight update only uses a single sampling instance. Therefore, it's a challenge to combine and coordinate the two transfer learning ways in a uniform framework. In this paper, inspired by the General Policy Iteration (GPI) in Dynamic Programming (DP) [26] where the GPI strategy is used to train the model based on Markov Decision Process (MDP). In DP training, the GPI strategy is designed to realize policy estimation and policy improvement, which are the mutual basis of self-value calculation.

Similar to the DP training, the online update and offline update are also based on mutual optimization results, i.e., the feature extraction (offline) and the label prediction (online) in DA are similar to the strategy improvement and the policy estimation in GPI, respectively. Thus, the whole transfer learning can be divided into two stages: offline stage (training with GPI strategy) and positioning stage (online learning and prediction).

In offline learning, the DANN is used to realize DA using labeled data of source domain and unlabeled data of target domain, it coincides with fingerprint adaptation of a positioning system where only the old labeled fingerprints and newly collected unlabeled fingerprints are available.

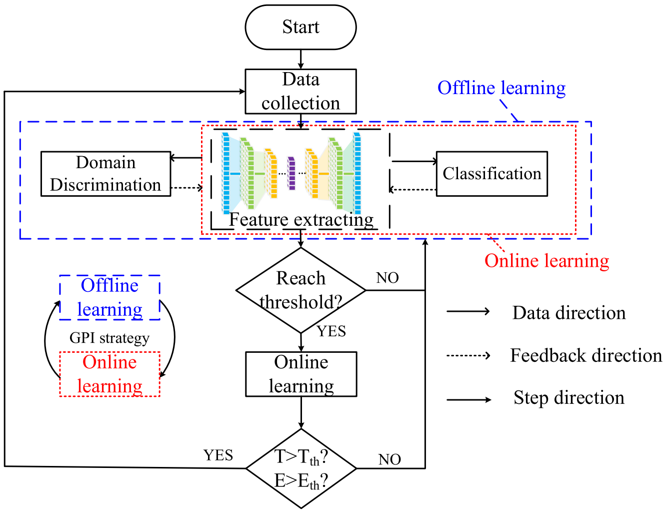


Fig. 4. The training flowchart of proposed framework.

In online learning, the PA algorithm is applied to fine-tune model, where the data is sent to training model in sequential manner. This characteristic of PA algorithm is compatible with the fingerprint recalibration of a localization system where the labeled fingerprints are collected one by one.

As shown in Fig. 4, the feature extraction part obviously involves both online learning and offline learning. Hence the GPI is applied to coordinate the relationship between DANN training and PA algorithm, the whole training sequence is described as follows:

- The data collection includes off-the-shelf labeled data in source domain and unlabeled data collected in target domain, it reflects that there is no laborious labeling process.
- Labeled data (source domain) and unlabeled data (target domain) are sent to feature extracting part for getting domain-invariant and prediction-discriminative features. After that, the GPI is used to adjust offline learning and online learning. Specifically, the weights of classification remain unchanged in offline learning stage (training domain discriminator and feature extractor), and the parameters of feature extracting cell are frozen for online learning of classification. Note that the labeled data in source domain is used in online learning.
- After reaching a preset accuracy (or threshold), the system fully switches to online learning with few labeled data in target domain. In this stage, the positioning system provides localization service and receives new labeled data (online learning) for calibrating the system. Therefore, this system has potential ability for anti-interference and adaptation in a dynamic environment. Once the positioning environment changes dramatically or the positioning error exceeds a given threshold, the whole system will return to the first step and retrain again.

As mentioned above, it can be seen that the whole system keeps continual training after offline learning, hence the proposed system has the potential for long-term stability compared with other single DA methods, moreover, the DA is embodied in feature representation, so that the final estimation (classification or regression) can give more reliable results based on the extracted features which are invariant and discriminative to the change of domain.

The transfer function of the feature extractor, domain discriminator and predictor (classification or regression), can be respectively expressed as $T_f(\cdot;\theta_f)$, $T_d(\cdot;\theta_d)$ and $T_p(\cdot;\theta_p)$; the parameter θ are a neural network weight and bias in our work. The predictor loss and domain discriminator loss under i-th input are respectively denoted by $l_p^i(\theta_f,\theta_p)$ and $l_d^i(\theta_f,\theta_d)$.

$$l_p^i(\theta_f,\theta_p) = l_p(T_p(T_f(\mathbf{x}_i;\theta_f);\theta_p),y_i), \tag{25}$$

$$l_d^i(\theta_f,\theta_d) = l_d(T_d(T_f(\mathbf{x}_i;\theta_f);\theta_d),d_i), \tag{26}$$

vector $\mathbf{x}_i$ is 1D input feature, where y_i and d_i are corresponding classification label and domain label.

According to reference [24], the offline DANN optimizing aims are

However, the classification optimizing mechanism of offline learning is different from that of online learning; thus, the former DANN classification optimizing equation $\hat{\theta}_p$ can be replaced with $l(\mathbf{w})$:

In order to fuse offline learning and online learning within a uniform framework, the GPI strategy is applied between offline learning and online learning, such that

Hence, in offline stage, the parameters updated by gradient descent are:

$$\theta_d \leftarrow \theta_d - \mu\lambda\frac{\partial l_d^i}{\partial\theta_d}, \tag{33}$$

$$\theta_f \leftarrow \theta_f - \mu\left(\frac{\partial l_p^i}{\partial\theta_f} - \lambda\frac{\partial l_d^i}{\partial\theta_f}\right). \tag{34}$$

In online stage, the parameters are updated by PA algorithm:

$$\mathbf{w}_j \leftarrow \mathbf{w}_j - \tau_j\left(1-\Psi_{\{j=y_i\}}\right)\mathbf{x}_i + \Psi_{\{j=y_i\}}\sum_{j\neq y_i}\tau_j\mathbf{x}_i, \tag{35}$$

where $l(\mathbf{w}^{(i)}) = l_p^i(\hat{\theta}_f,\theta_p)$, i.e., the parameter θ_p denotes the weight matrix $\mathbf{w}$.

We give the structure of the network and algorithm in Fig. 5 and Table II, respectively, where it should be noted that there is a gradient reversal layer between feature extraction layer and domain classifier in Fig. 5, the gradient reversal layer reverses the gradient (multiplying a negative scalar) in the error backpropagation and keep the input unchanged in the forward propagation. Thresholds T_{th} and E_{th} in Table II represent the time threshold and error threshold of online learning, respectively. Online learning is not suitable for long-term positioning, and is only used for model tuning in a slightly changing environment. Hence the time threshold T_{th} and error threshold E_{th} are needed to change online learning mode to offline learning mode for keeping positioning accuracy. The determination of T_{th} and E_{th} needs to be set according to the specific environment and positioning requirements: the more complex the environment, the smaller the T_{th}; the higher the positioning accuracy requirement, the smaller the E_{th}.

TABLE II
THE TRAINING ALGORITHM

```
Algorithm: The Training Procedure of the Proposed Framework
Input: samples: $S=\{\mathbf{x}_i,\mathbf{y}_i\}_{i=1}^{n}\sim(\mathbf{D}_S)^n$ , $T=\{\mathbf{x}_i\}_{i=n+1}^{n+m}\sim(\mathbf{D}_T^X)^m$ ,learning rate: $\mu$ ,hyperparameter: $\lambda$ ,batch size: $b$
Random Initialize: $\theta_p$ , $\theta_d$ , $\theta_f$
while stopping criterion is not met do
  for epoch $v\in[0,V]$ do
    for each minibatch $mb\in[0,(m+n)/b]$ do #offline learning
      $\hat{y}=T_p(T_f(\mathbf{x};\theta_f);\hat{\theta}_p)$
      $\hat{d}=T_d(T_f(\mathbf{x};\theta_f);\theta_d)$
      source domain loss: $l_p(\theta_f,\hat{\theta}_p)=l_p(\hat{y},y)$
      target domain loss: $l_d(\theta_f,\theta_d)=l_d(\hat{d},d)$
      $\theta_d\leftarrow\theta_d-\mu\lambda\frac{\partial l_d}{\partial\theta_d}$
      $\theta_f\leftarrow\theta_f-\mu\left(\frac{\partial l_p}{\partial\theta_f}-\lambda\frac{\partial l_d}{\partial\theta_f}\right)$
    end for
    for each sample $(\mathbf{x}_i,y_i)$ in $S$ do   #online learning
      while $e\in Y\setminus\{y_i\}$ do
        $l_e=\max\left[0,1-\left(\mathbf{w}_{y_i}\cdot\mathbf{x}_i-\mathbf{w}_e\cdot\mathbf{x}_i\right)\right]$
      end while
        sort $\{l_e|e\in Y\setminus y_i\}$ in descending order as $\{l_{e(1)},\ldots,l_{e(K-1)}\}$
        $k=1$ , $SP=\varnothing$
      while $\sum_{i=1}^{k-1}l_{e(i)}<\min\left(\frac{(k+1)C}{\|\mathbf{x}_j\|^2}-l_{e(k)},kl_{e(k)}\right)$ do
        $SP=S\bigcup\{e(k)\}$ , $k=k+1$ #support vector
      end while
      while $r\in S$ do
        $\tau_r=\|\mathbf{x}_i\|^{-2}\max\left[0,l_r-\max\left(\sum_{u\in S}\frac{l_u}{|S|}+\frac{C}{|S|}\|\mathbf{x}_i\|^2,\sum_{u\in S}\frac{l_u}{|S|+1}\right)\right]$
        $\mathbf{w}_{y_i}\leftarrow\mathbf{w}_{y_i}+\tau_r\mathbf{x}_i$
        $\mathbf{w}_r\leftarrow\mathbf{w}_r+\tau_r\mathbf{x}_i$
      end while
    end for
  end for
end while
while true do # turning into fully online learning
  receiving online training data $(\mathbf{x}_i,y_i)$ or predicting data $\mathbf{x}_i$
  if training data is true then
    call online learning module and input $(\mathbf{x}_i,y_i)$
  else
    give predicted value: $\hat{y}=T_p(T_f(\mathbf{x};\theta_f);\theta_p)$
end if
if $T>T_{th}$ or $E>E_{th}$ then
  break
end if
end while
restart this algorithm
```

V. Experiment Evaluations

In this section, we will firstly give the basic experiment environment and condition of data preprocessing. Then, series

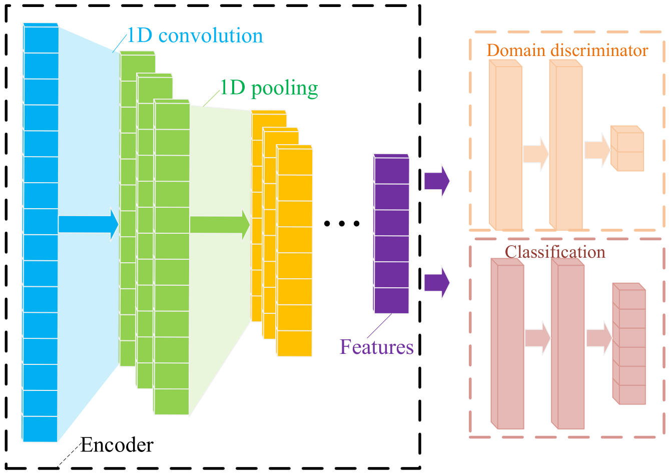


Fig. 5. The structure of the localization framework.

of experiment are conducted to verify the effectiveness and robustness of our proposed algorithm.

A. Experiment Preparation

To verify the superiority of proposed algorithm, various experiments are conducted in five scenes, as shown in Fig. 6 including an office room (E1), a small-size office room (E2), a hybrid experiment room (E3), a meeting room (E4) and a semi-open corridor (E5). In Fig. 6, the layout of the true environment means the stable furnishings and decorations that remain unchanged (location and state) in all experiments. In each experiment, an ASUS laptop equipped with 5300 NIC is used as signal receiver (on a table with a height of 0.83m), where the CSI tool [50] is running under Ubuntu 14.0 operating system to capture signals; Xiaomi R4CM router (on a table with a height of 0.75m) with 2.4GHz frequency band and power of 5W is used as the transmitter to send signals to receiver. The sampling frequency is 5 Hz which means that the interval of adjacent package is 200ms. Each point of fingerprint collection in source domain is sampled for 5 minutes (labeled), and each point of fingerprint collection in the target domain is sampled for half a minute (unlabeled). The format of received CSI in each package is 2 (transmitting antenna) ×3 (receiving antenna) ×30 (subcarrier). All experimental data are processed on a computer with an Intel Core i5-10400F CPU, 16 GB RAM and NVIDIA GeForce RTX 2060. The ranging device used in all tests is DELIXI DB50 laser rangefinder.

As listed in Table III, five scenes with different conditions are designed to conduct a series of experiments. In scene E1, the state of source domain is closing door (E1_S), and the state of target domain is opening windows and adding baffle (E1_T), it is designed to evaluate the adaptive ability of proposed method in minor change environment. The difference between source domain (E2_S) and target domain (E2_T) in E2 is whether there is the pedestrian (two pedestrians) and baffle

$$\hat{\theta}_d = \arg\max_{\theta_d}\left[\frac{1}{n}\sum_{i=1}^{n} l_p^i(\hat{\theta}_f,\hat{\theta}_p) - \lambda\left(\frac{1}{n}\sum_{i=1}^{n} l_d^i(\hat{\theta}_f,\theta_d) + \frac{1}{m}\sum_{i=n+1}^{n+m} l_d^i(\hat{\theta}_f,\theta_d)\right)\right], \tag{27}$$

$$(\hat{\theta}_f,\hat{\theta}_p) = \arg\min_{\theta_f,\theta_p}\left[\frac{1}{n}\sum_{i=1}^{n} l_p^i(\theta_f,\theta_p) - \lambda\left(\frac{1}{n}\sum_{i=1}^{n} l_d^i(\theta_f,\hat{\theta}_d) + \frac{1}{m}\sum_{i=n+1}^{n+m} l_d^i(\theta_f,\hat{\theta}_d)\right)\right], \tag{28}$$

$$\begin{aligned}\hat{\theta}_p &= \arg\min_{\theta_p}\left[\frac{1}{n}\sum_{i=1}^{n} l_p^i(\hat{\theta}_f,\theta_p) - \lambda\left(\frac{1}{n}\sum_{i=1}^{n} l_d^i(\hat{\theta}_f,\hat{\theta}_d) + \frac{1}{m}\sum_{i=n+1}^{n+m} l_d^i(\hat{\theta}_f,\hat{\theta}_d)\right)\right]\\ &= \arg\min_{\theta_p}\frac{1}{n}\sum_{i=1}^{n} l_p^i(\hat{\theta}_f,\theta_p),\end{aligned} \tag{29}$$

$$l(\mathbf{w}^{(i)}) = \max\left[0, 1-\left(\mathbf{w}_{y_i}^{(i)}\cdot\mathbf{x}_i - \max_{j\neq y_i}\mathbf{w}_j^{(i)}\cdot\mathbf{x}_i\right)\right], \tag{30}$$

$$\begin{cases}\hat{\theta}_d = \arg\max_{\theta_d}\left[\frac{1}{n}\sum_{i=1}^{n} l_p^i(\hat{\theta}_f,\hat{\theta}_p) - \lambda\left(\frac{1}{n}\sum_{i=1}^{n} l_d^i(\hat{\theta}_f,\theta_d) + \frac{1}{m}\sum_{i=n+1}^{n+m} l_d^i(\hat{\theta}_f,\theta_d)\right)\right]\\ = \arg\max_{\theta_d}\left[-\lambda\left(\frac{1}{n}\sum_{i=1}^{n} l_d^i(\hat{\theta}_f,\theta_d) + \frac{1}{m}\sum_{i=n+1}^{n+m} l_d^i(\hat{\theta}_f,\theta_d)\right)\right]\\ \hat{\theta}_f = \arg\min_{\theta_f}\left[\frac{1}{n}\sum_{i=1}^{n} l_p^i(\theta_f,\hat{\theta}_p) - \lambda\left(\frac{1}{n}\sum_{i=1}^{n} l_d^i(\theta_f,\hat{\theta}_d) + \frac{1}{m}\sum_{i=n+1}^{n+m} l_d^i(\theta_f,\hat{\theta}_d)\right)\right],\end{cases} \tag{31}$$

$$l(\mathbf{w}^{(i)}) = \max\left[0, 1-\left(\mathbf{w}_{y_i}^{(i)}\cdot\mathbf{x}_j - \max_{j\neq y_i}\mathbf{w}_j^{(i)}\cdot\mathbf{x}_i\right)\right]. \tag{32}$$

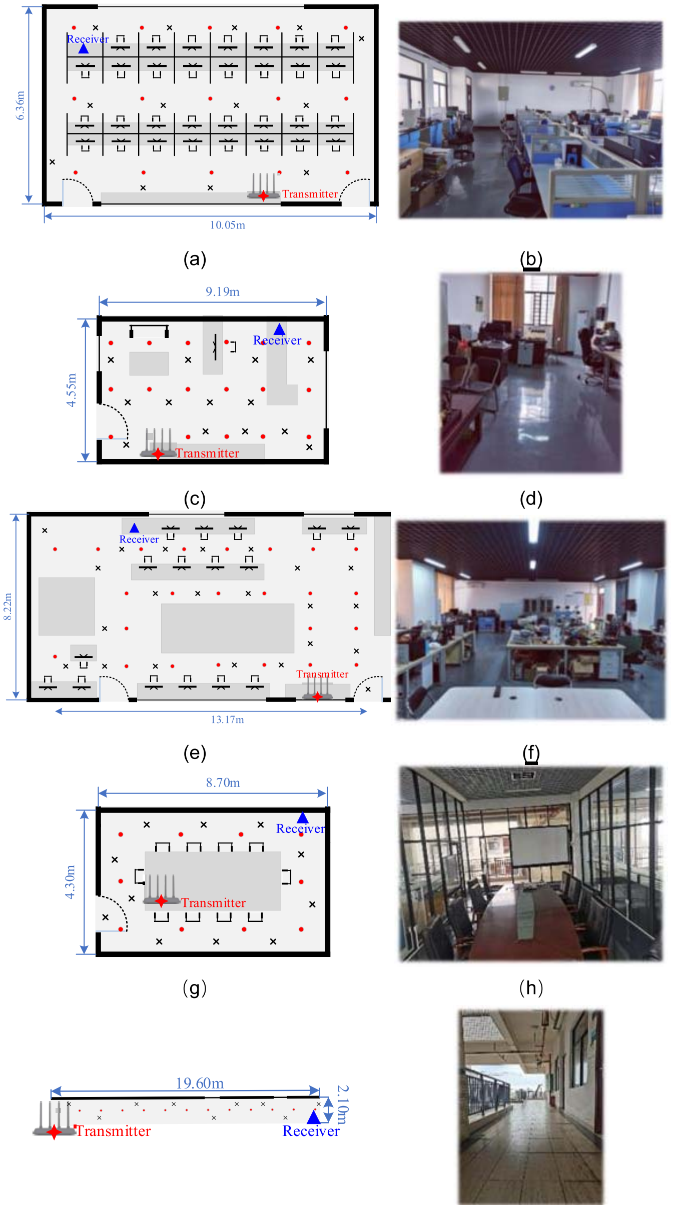


Fig. 6. The left column is the environment layout, and the right column is the real environment. From top to bottom are E1, E2, E3, E4 and E5. The red quadrangle is the location of the transmitter and the blue triangle is the location of the receiver. The red dots are reference points. The black crosses are test points.

TABLE III
THE CONDITIONS OF SOURCE DOMAIN AND TARGET DOMAIN

Testbed	Size (m^2)	Change description
E1: office_1	10.05×6.36	1. Open door and window 2. Add baffle
E2: office_2	9.19×4.50	1. Pedestrian interference 2. Add baffle
E3: lab	13.17×8.22	1. Change Wi-Fi equipment 2. Change device location
E4: meeting room	8.70×4.30	1. Block transmitter 2. Block receiver
E5: corridor	19.60×2.10	1. pedestrian interference 2. Move metal cabinet

interference (a whiteboard). Specifically, in the target domain of E2, one student stands almost motionless, the other one wanders randomly, and the whiteboard is placed between the signal receiver and the transmitter. The purposed of E2 design is to test the localization performance of proposed method in true pedestrian interference environment. To model different environment fingerprint transfer, the source domain and target domain in E3 use different Wi-Fi devices and anchor positions, i.e. the Wi-Fi device is changed into TP-LINK AC1200 router with 2.4GHz frequency band and power of 7.65W and placed on the ground (same plane coordinates) in target domain. The testbed E4 is a small meeting room constructed with tempered glass different from traditional concrete wall, and we collected the CSI fingerprints in E4 before and after two people blocking transmitter and receiver (E4_S and E4_T), which is realistic scenario in daily life. We also collected fingerprint information of corridor (E5) under three pedestrian interference (wandering randomly) and changing the location of metal cabinets, taking corridor as testbed is necessary because it is a part of indoor space and it is neglected in most studies. Besides, the corridor is a semi-open space (as shown in Fig. 6) different from standard room, which is very representative. Thus corridor (E5) is chosen as testbed in our research. The dynamic interference variables of corridor are mostly pedestrian interference, movement of furniture or facilities (such as metal fire cabinet), etc. Therefore, we mainly consider the interference of people and metal fire cabinet in corridor.

For fingerprint collection, we firstly collect an appropriate number of labeled fingerprints at the reference points on each test platform to build the source database (five minutes per point). After that, we change the condition of environment according to Table III, and then we collect the unlabeled fingerprints and the labeled fingerprints at the reference points and test points (half a minute for each point), respectively. At last, the collected fingerprints (labeled data in source domain and unlabeled data target domain) are sent to train our proposed model. After finishing training, the performance of the algorithm is verified using the labeled fingerprints collected at the test points. The label of fingerprint adopts the nearest principle, i.e., the fingerprint collection points are classified as their nearest reference point.

B. Basic Performance Test

In this part, we will verify the superiority of proposed algorithm by comparing with several benchmark methods in three scenes (E1-3), the target domain consists of unlabeled fingerprints, while the source domain consists of labeled data. All the tests are performed using unlabeled target data unless otherwise stated.

We firstly give the training processes of proposed algorithm, including classification accuracy and loss of test in E1. As shown in Fig. 7, in the early training stage, the classification accuracy in target domain fluctuates sharply and the classification accuracy in source domain is unstable. However, once the accuracy of target classification is stable, the source domain classification will remain stable for a long time; this

TABLE IV
THE COMPARISONS OF POSITIONING PERFORMANCE WITH BASIC METHODS

Item	Scenario	DANN	DANN-PA cascading	PA	SVC	KNN	Proposed
Classification right (%)	E1	25.7%	30.3%	27.6%	21.9%	21.0%	100%
	E2	38.2%	50.7%	35.6%	20.2%	22.7%	98.2%
	E3	38.8%	33.3%	20.8%	22.4%	25.5%	84.9%
Mean error (m)	E1	2.13	1.98	2.15	2.25	2.28	0.77
	E2	2.53	1.93	2.27	2.41	3.22	0.59
	E3	3.20	4.05	5.01	4.05	3.87	1.10
Std of error	E1	1.24	1.26	1.55	1.36	1.21	0.29
	E2	2.49	1.73	1.75	1.48	2.50	0.34
	E3	3.11	3.40	3.27	3.00	3.10	1.39

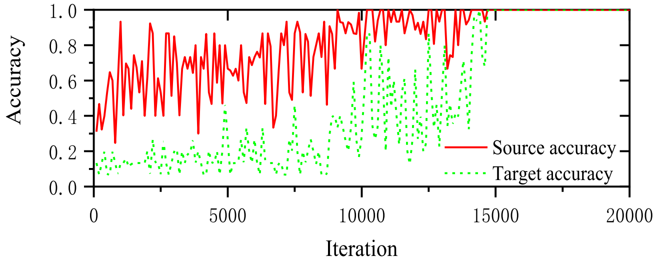
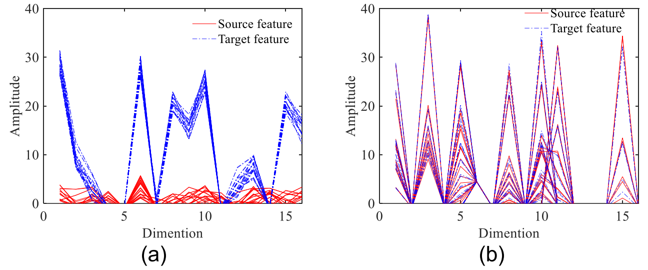


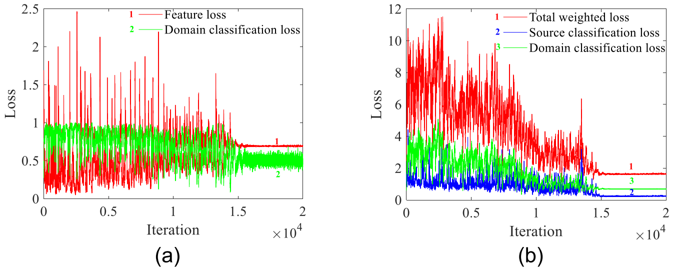
Fig. 7. The classification accuracy of proposed algorithm in source domain and target domain of E1.



(a)

(b)

Fig. 9. The features of source domain and target domain provided by 1D-CAE, each curve presents features of one location. (a) 1D-CAE output features before training; (b) 1D-CAE output features after training.



(a)

(b)

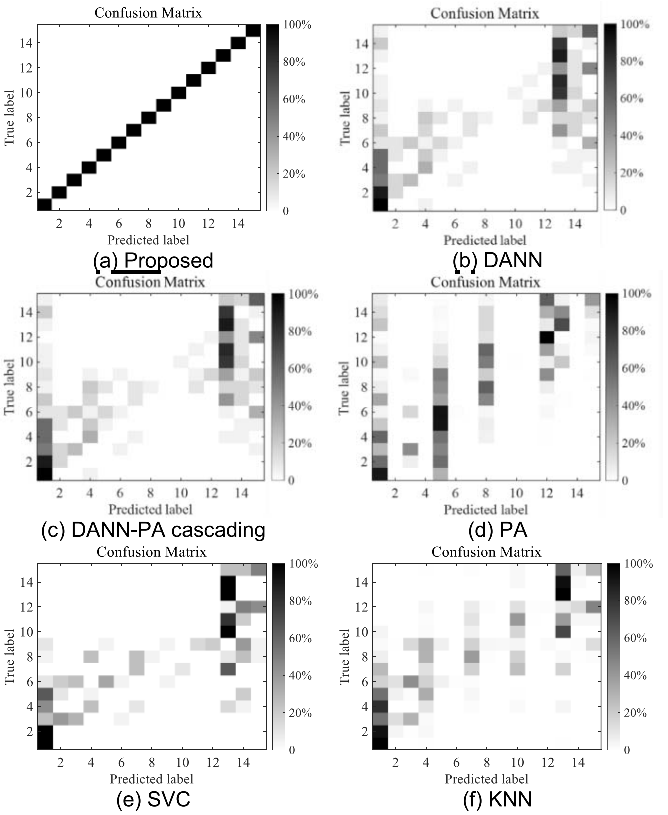
Fig. 8. The losses of proposed algorithm in E1. (a) The loss of domain adversarial stage of proposed algorithm; (b) The loss in classification training stage.

feature can be adopted as a training stop indicator, because we cannot get classification accuracy of target domain with only unlabeled data in training stage. Fig. 8 depicts the change of loss in different training stages, it shows that all losses will gradually stabilize as the accuracy of the classification in the target domain reaches the maximum value. The changes of loss and accuracy in E2 and E3 DA is like that of E1, thus they are omitted in latter tests.

The Fig. 9 shows that the features from both source domain and target domain will be mapped into same feature space using proposed algorithm, thus better prediction results will be achieved after mapping.

The proposed algorithm is compared with several benchmark methods, including DANN, PA (*C=1*), Support Vector Machine (SVM, *C=1, decision_function_shape='ovr', kernel='rbf'*), KNN (*k=5*), DANN-PA cascading (Same configuration as DANN and PA). The DANN-PA cascading strategy is direct combination of two algorithm where the PA uses the features provided by DANN to predict location, which is similar with loose coupling in sensor fusion. However, in our proposed algorithm, the PA algorithm is embedded in DANN, and the GPI strategy is used for adversarial training between two algorithms. The localization results in E1 are shown in Fig. 10 (confusion matrix) and the corresponding statistical data are listed in Table IV.

As shown in Fig. 10, the proposed algorithm has better localization results compared with other benchmark methods. Though our proposed algorithm is related with DANN and PA, the proposed algorithm outperforms DANN and PA in positioning application. The proposed algorithm fully exploits DANN and PA, where the abilities of feature adaptation and classification are considerably improved with the help of GPI. The experiment results show that the proposed algorithm is not simply cascading DANN with PA. Different from DANN-PA cascading strategy, the proposed algorithm is not sequentially using two basic algorithms, but fusing PA into DANN training with GPI strategy, as shown in algorithm 1. Fig. 10 (b) and (c) show that cascading method of DANN and PA may reduce the positioning accuracy compared with single scheme. Fig. 10 also indicates that traditional non-DA methods (Fig. 10 (d)-(f)) do not perform as well as DA methods (Fig. 10 (a)-(c)) in dynamic environment. To quantitatively evaluate localization performance, the basic performance indicators are calculated, including classification accuracy, mean positioning errors and Standard deviation (Std) of positioning errors, as listed in Table IV. The localization error is represented by the distance from the actual position to the geometric center of the prediction area, i.e., the positioning mean error is the average of the distance between the predicted grid center points (reference points) and the actual locations. The Standard Deviation (Std) is used to reflect the stability of the positioning system.

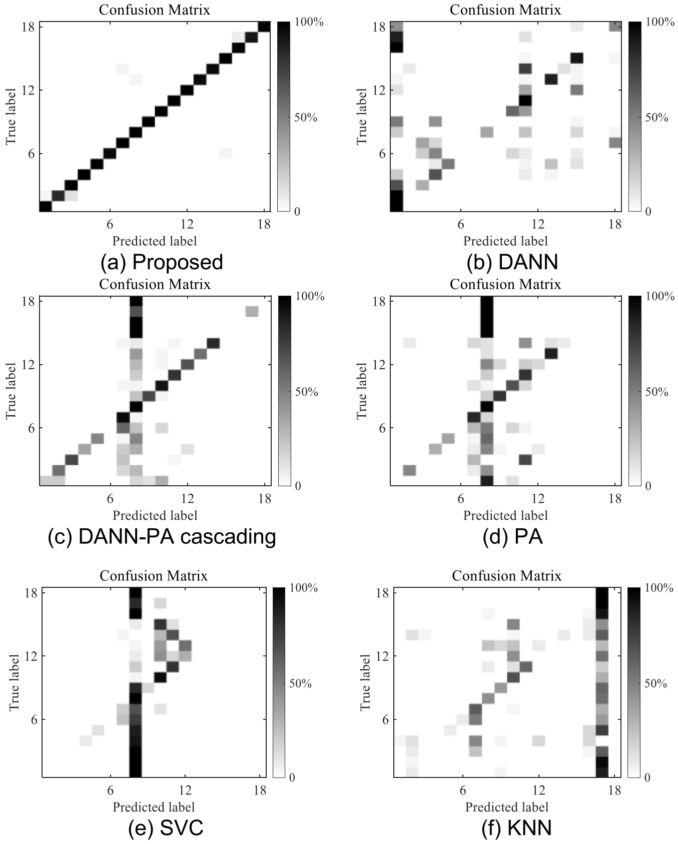


(a) Proposed (b) DANN

(c) DANN-PA cascading (d) PA

(e) SVC (f) KNN

Fig. 10. Prediction results with different methods in E1. Source domain is E1_S, target domain is E1_T.



(a) Proposed (b) DANN

(c) DANN-PA cascading (d) PA

(e) SVC (f) KNN

Fig. 11. Prediction results with different methods in E2. Source domain is E2_S, target domain is E2_T.

As shown in Fig. 10-12 and Table IV, due to the interference of pedestrian and baffle, the features of fingerprint vary with time periodically, thus, the localization system needs offline feature transfer (eliminate baffle effect and mitigate pedestrian effect) and online fine tuning. Compared with E1, the interference in E2 consists of stable change (baffle), dynamic change (pedestrian walking). Therefore, the traditional positioning method without DA cannot provide accurate results. Though, DANN provide offline feature transfer, it cannot realize online fine-tuning; PA adjusts positioning network using online learning, but it is hard to completely eliminate baffle and pedestrian effects; the DANN-PA cascading strategy cannot solve these problems, because DANN and PA algorithm are utilized separately to realize feature transfer without any cooperation, which may lead to mutual restraint between DANN and PA algorithm. As shown in Fig. 12, the fingerprints are fully changed in E3_T, hence the traditional classification methods give chaotic results (Fig. 12 (e) and (f)). In contrast, the adaptive methods have better predicted results. As listed in Table IV, the proposed algorithm has the highest accuracy, small mean error and Std.

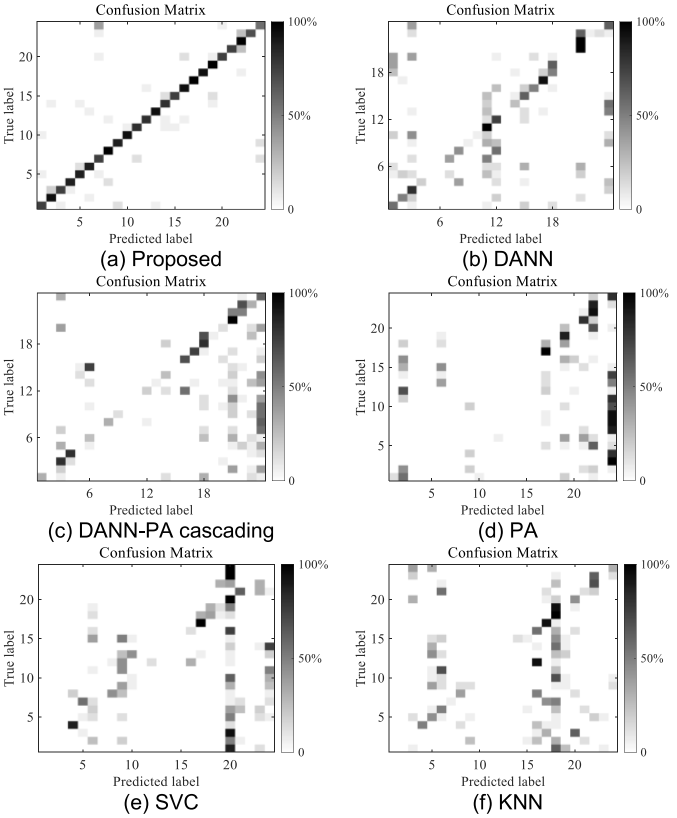
C. Comparison Existing Works

The proposed adaption algorithm is compared with several state-of-the-art indoor positioning works: Zheng *et al.* adopt the phase autoencoder (P-**AE**) and the amplitude autoencoder (A-**AE**) to calibrate real-time CSI measurements [22], thus AE can realize feature transforming i.e., mapping target and source features into common feature space. The **AdapLoc** [23] based on semantic alignment is proposed to realize DA in dynamic environment, it should be noted that the AdapLoc is designed for regression mission in original work, thus we replace the last regression layer of AdapLoc with classification layer in this comparison. Similarly, a modified **TrAdaBoost** is proposed to improve positioning performance in dynamic environment [40]. The proposed algorithm is also compared with two deep feature extraction schemes with good performance in indoor positioning: Wang *et al*. designed a deep residual sharing neural network (**ResNet**) to extract deep and stable characteristics of CSI fingerprint through bi-modal CSI tensor [51]. Liu *et al*. constructed a deep neural network consisted of local connection (**LC**) and full connection layers to extract robust feature expressions of CSI fingerprints [52].

1) Basic Performance Comparison: The performances of the mentioned works tested in five environments (E1-E5) are shown in Fig. 13-17 and listed in Table V. It shows that all the methods have good performance in E1 (except LC), because the change of E1 is relatively stable; but for LC, the baffle hinders its ability to extract adjacent channel features, i.e. it may be limited by Non-Line-of-Sight (NLoS). The target environment of E2 is harsh and slightly fluctuant due to stationary baffle and wandering people, thus, the localization results are worse compared with E1. In E3, the location and type of Wi-Fi devices are changed, which reduce the fingerprint correlation between source environment and

TABLE V
THE COMPARISONS OF POSITIONING PERFORMANCE WITH RECENT WORKS

Item	Scenario	P-AE+ A-AE [20]	AdapLoc [21]	TrAdaBoost [31]	ResNet [51]	LC [52]	proposed
Classification right (%)	E1	75.6%	100%	100%	54.6%	27.5%	100%
	E2	67.8%	87.3%	86.7%	33.4%	20.0%	98.2%
	E3	43.0%	75.5%	72.9%	27.7%	13.4%	84.9%
	E4	61.1%	93.3%	97.3%	28.9%	30.0%	100%
	E5	70.1%	91.7%	98.1%	68.4%	41.3%	100%
Mean error (m)	E1	1.66	0.77	0.77	2.01	3.05	0.77
	E2	1.31	0.70	0.71	1.96	2.44	0.59
	E3	2.70	1.31	1.35	3.50	4.16	1.11
	E4	1.02	0.59	0.59	1.80	1.82	0.54
	E5	2.33	0.84	0.65	2.41	4.06	0.54
Std of error	E1	2.00	0.29	0.29	2.21	2.57	0.29
	E2	1.71	0.60	0.68	1.65	1.85	0.34
	E3	3.16	1.45	1.66	3.06	2.67	1.39
	E4	0.95	0.38	0.42	1.40	1.51	0.21
	E5	3.99	1.53	0.84	3.42	4.59	0.21



(a) Proposed

(b) DANN

(c) DANN-PA cascading

(d) PA

(e) SVC

(f) KNN

Fig. 12. Prediction results with different methods in E3. Source domain is E3_S, target domain is E3_T.

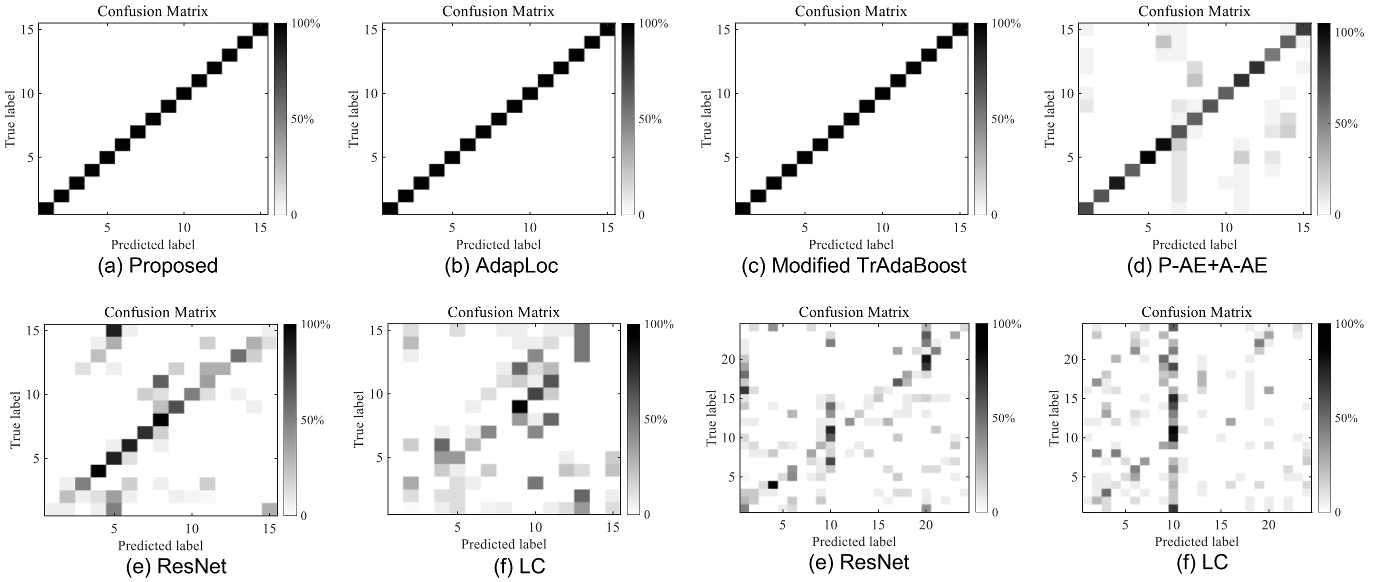
target environment, consequently, the positioning results of all methods are poor. Compared with E1-E3, the testbeds E4 and E5 have less classification categories. Hence, the non-adaptation methods ResNet and LC has better localization results. Although schemes ResNet and LC have higher accuracy in relatively static environments (e.g., E5), they cannot have the same performance in complex dynamic environments. Because the essential ability of schemes ResNet and LC is powerful feature extraction, it can extract robust features in a mild environment with acceptable environmental ranges. Once the environment changes drastically, it is hard to find common robust features between source domain and target domain only by enhancing the feature extraction capability of the localization system.

Figures 13-17 and Table V indicate that all listed works have better positioning results with slight environmental change (E1), while dynamic interference (E2) and uncorrelated change (E3) are irregular which is hard to be completely eliminated. Besides, the experiments in E4 and E5 verify that the traditional deep learning-based positioning system is only suitable for the relatively stable environment with few positioning categories. All experimental results show that our proposed method has best performance compared with listed similar works, the essence of our method is deeper characteristics extraction and fusion of online and offline transfer.

In order to more intuitively show the performance of various algorithms, the Cumulative Distribution Function (CDF) of classification accuracy is adopted to compare the positioning results. As shown in Fig. 18, the CDF of our scheme is always on the far right and rises the fastest, that is, our scheme has the highest positioning accuracy.

2) Comparison in Incrementally Changing Environment: To quantitively compare the performance between different methods under different degrees of interference, we incrementally add interference factor in each target domain, and the corresponding fingerprints are collected for DA and location estimation. In this comparison test, testbed E5 is chosen as test environment and the number of people is set as interference variable. Then we can get the performance of various approaches under different numbers of people. In order to control other variable interference, this test is conducted in same day.

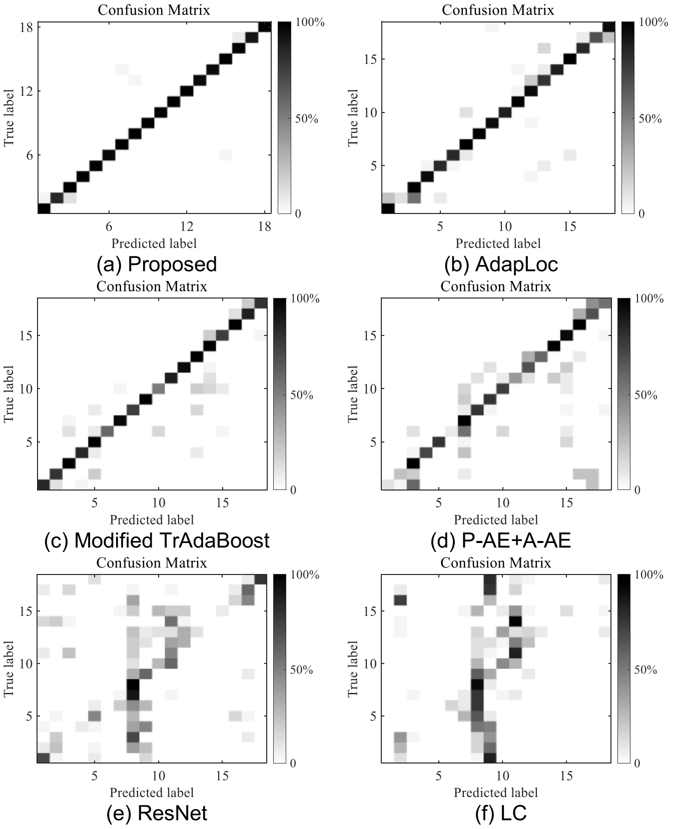
The comparison results are shown in Fig. 19, it can be seen that our proposed scheme is more stable and has higher accuracy compared with other schemes, and if the labeled data accounts for 1/10, the prediction accuracy will reach 100%. In this comparison, the KNN acts as a no-adaptation



(a) Proposed (b) AdapLoc (c) Modified TrAdaBoost (d) P-AE+A-AE

(e) ResNet (f) LC (e) ResNet (f) LC

Fig. 13. Classification results in E1.



(a) Proposed (b) AdapLoc

(c) Modified TrAdaBoost (d) P-AE+A-AE

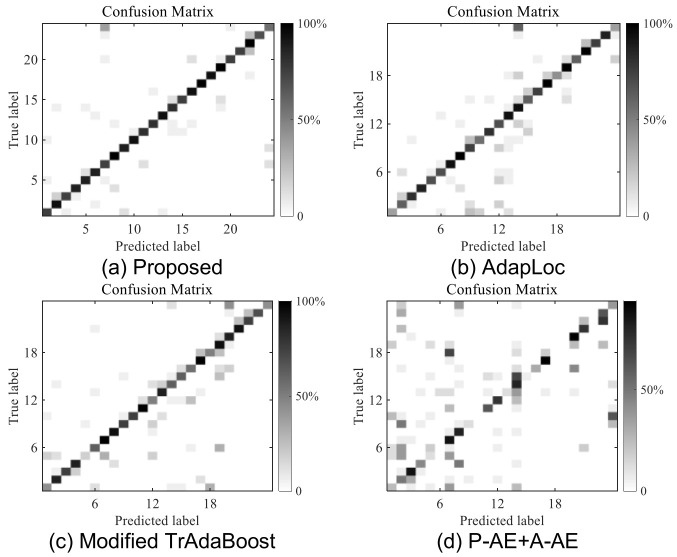
(e) ResNet (f) LC

Fig. 14. Classification results in E2.

benchmark method. The accuracy decline of KNN is not obvious after reaching 3 persons, this may be because this type of interference has nearly reached its limit, i.e., some signal transmission paths (Line of Sight) are almost completely closed after 3 persons.

D. Data Number Effect

Intuitively, the performance of transfer learning is related with environment type, number of fingerprints, fingerprint collection interval, localization granularity (classification interval) and so on. The effect of environment type on feature transfer is hard to qualitatively described. Hence, we will only discuss localization performance under different fingerprint number, fingerprint intervals and classification number.



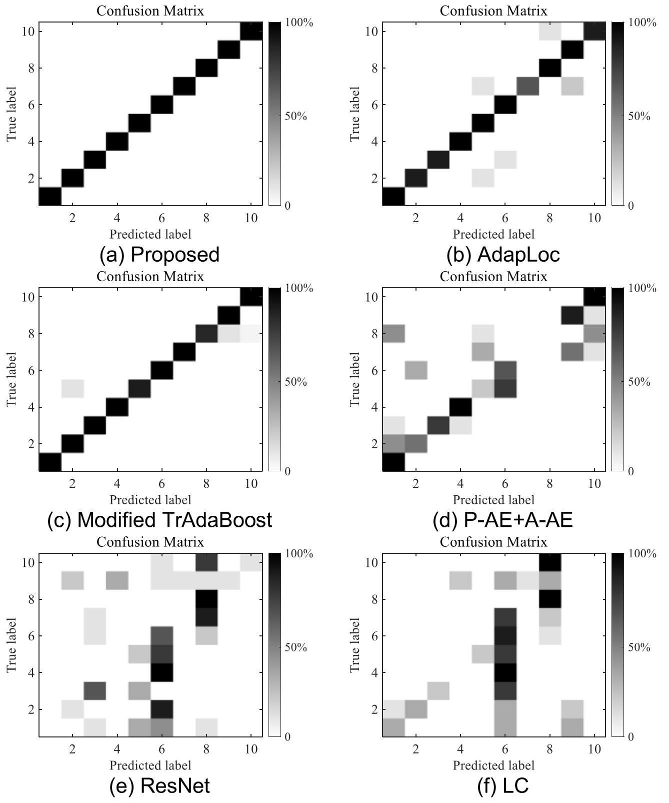
(a) Proposed (b) AdapLoc

(c) Modified TrAdaBoost (d) P-AE+A-AE

Fig. 15. Classification results in E3.

1) Fingerprint Number Effect: To quantitatively explore the relationship between the sample number of target domain and the positioning accuracy, several tests are conducted under different ratio between the source fingerprints and the target domain fingerprints. In this test, the number of source fingerprint is constant.

Table VI is the variation of the positioning accuracy with the ratio of the target fingerprint to the source fingerprint, it shows that the proposed method can achieve high positioning accuracy with less unlabeled fingerprints, thus it can alleviate fingerprint collection problem when the environment changes. As listed in Table VI, though increasing fingerprint number in target source can improve positioning performance, the



(a) Proposed

(b) AdapLoc

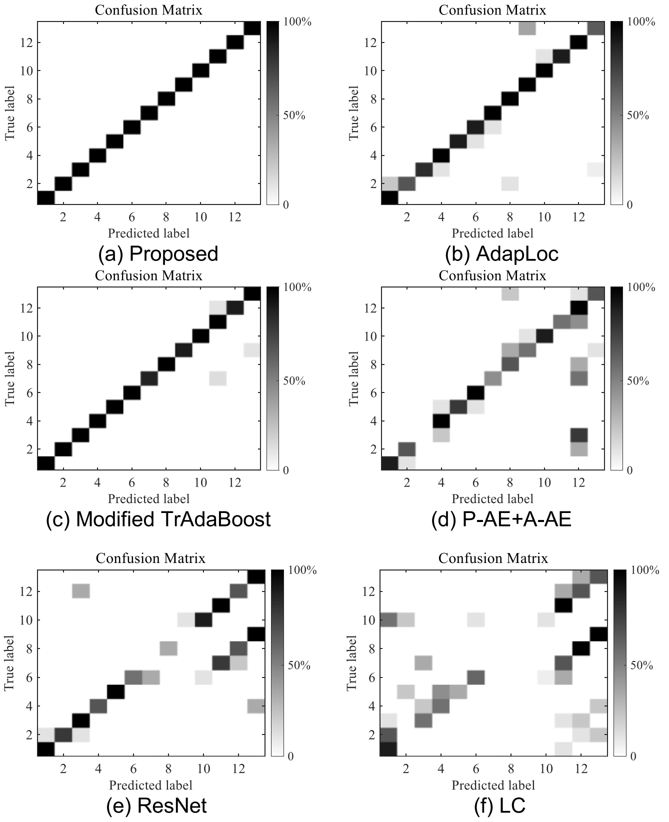
(c) Modified TrAdaBoost

(d) P-AE+A-AE

(e) ResNet

(f) LC

Fig. 16. Classification results in E4.



(a) Proposed

(b) AdapLoc

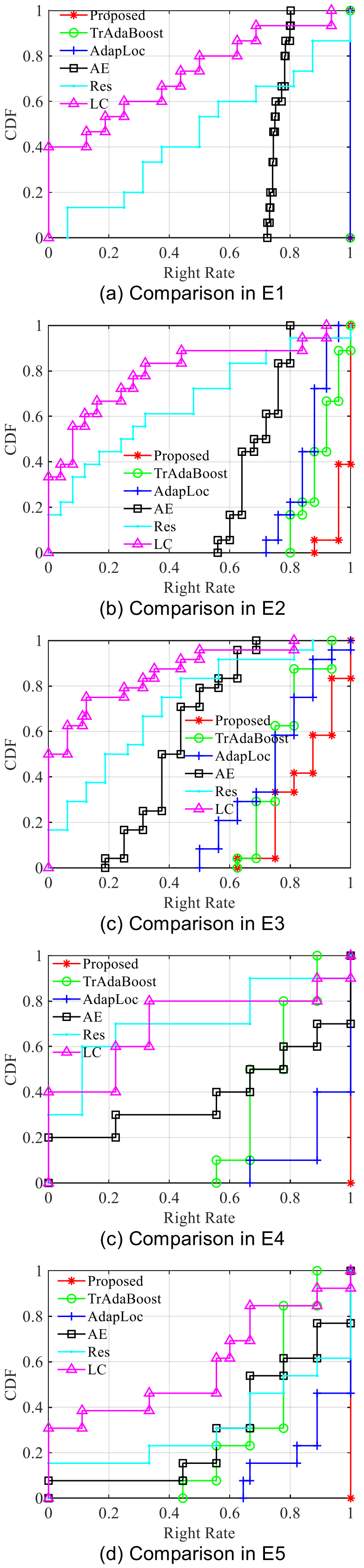
(c) Modified TrAdaBoost

(d) P-AE+A-AE

(e) ResNet

(f) LC

Fig. 17. Classification results in E5.



(a) Comparison in E1

(b) Comparison in E2

(c) Comparison in E3

(c) Comparison in E4

(d) Comparison in E5

Fig. 18. The CDF of prediction accuracy of each algorithm.

classification accuracy cannot infinitely increase with the number of fingerprints. Conversely, large number of fingerprints may reduce the accuracy and increase training cost due to noise interference in collected data.

2) Fingerprint Interval Effect: Fingerprint interval has intrinsic effect on localization performance. To test the effect of fingerprint interval, we change the fingerprint interval while the

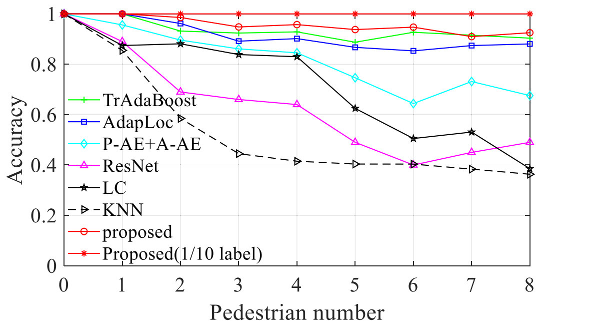


Fig. 19. Classification results in incrementally changing environment.

TABLE VI
THE LOCALIZATION CLASSIFICATION ACCURACY UNDER DIFFERENT TARGET /SOURCE FINGERPRINT RATIO

Ratio	0.2	0.5	1	1.5	2
Classification accuracy	81.6%	82.5%	84.9%	89.6%	87.8%

TABLE VII
THE RELATIONSHIP BETWEEN ACCURACY AND FINGERPRINT INTERVAL

Interval	2	1	0.5	0.25
Classification accuracy	54.5%	75.91%	84.9%	81.8%

TABLE VIII
THE CLASSIFICATION ACCURACY UNDER DIFFERENT RATIO OF LABELED DATA IN TARGET DOMAIN

Ratio	0	1/20	1/10	1/5	1/2	1
E1	100%	100%	100%	100%	100%	100%
E2	98.2%	100%	100%	100%	100%	100%
E3	84.9%	100%	100%	100%	100%	100%

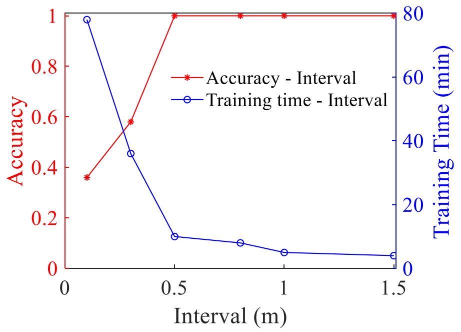


Fig. 20. The effects of interval on classification accuracy and training time.

TABLE IX
1D-CAE WITH DIFFERENT OUTPUT FEATURE DIMENSION

1D-CAE Structure	Mean Classification Accuracy	Mean Time Cost
64-32-4-32-64	54.6%	2h 23min 19s
64-32-8-32-64	84.9%	2h 55min 5s
64-32-12-32-64	83.6%	3h 32min 27s
64-32-16-32-64	79.5%	4h 8min 43s

TABLE X
1D-CAE WITH DIFFERENT LAYER NUMBER

1D-CAE Structure	Mean Classification Accuracy	Mean Time Cost
32-8-32	74.4%	2h 38min 34s
64-32-8-32-64	84.9%	2h 55min 5s
128-64-32-8-32-64-128	85.8%	4h 49s
256-128-64-32-8-32-64-128-256	79.5%	4h 8min 43s

TABLE XI
ONLINE REAL-TIME PERFORMANCE

	Average Calibrating Time (ms)	Average Predicting Time (ms)
Time Cost	60.6	65.8

locations of fingerprint collection points are equally distributed in the interesting area, and the interval between classification points is constant.

As is listed in Table VII, the classification accuracy does not always increase as the interval decreases due to noise interference. The noise will exceed the difference between the fingerprint of adjacent points when fingerprint interval is small enough, hence proper fingerprint collection interval needs to be determined before fingerprint collection.

3) Classification Interval Effect: To qualitatively test localization performance under different classification numbers, the localization granularity (classification interval) is used to represent the change of classification number, because the more classification categories, the smaller the granularity. In this test, the target/source fingerprint number ratio is 1 and test scenario is E1.

As shown in Fig. 20, the classification accuracy increases as the granularity increases; conversely, the offline training time decreases as the granularity increases. It reveals that position-specific fingerprint is harder to judge as the granularity decreases, once the granularity is decreased to a certain extent, the signal noise will annihilate the fingerprint difference between adjacent positions, which will result in classification error and longtime training.

4) Hybrid Database With Labeled Data and Unlabeled Data: To simplify the discussion, we set the total number of target fingerprints to be the same as the source domain and get the localization results under different ratio between labeled data and unlabeled data in target domain. The test results are listed in Table VIII, it reveals that only a small amount of labeled data can considerably improve the accuracy of classification, which means that our scheme can reduce the time consuming and labor cost in fingerprint collection.

E. Structure Effect and Real-Time Performance

Feature extraction is the basis of transfer learning, which nearly determines the efficiency of feature transfer in offline stage, moreover, the position estimation results mostly rely on the feature extracted by 1D-CAE. Therefore, the effects of parameter in 1D-CAE will be discussed in this part. After that, the real-time performance of online localization and recalibration is also tested. These tests are performed in E3.

As listed in Table IX, the best output dimension is 8 with highest classification and less training time. Although deeper

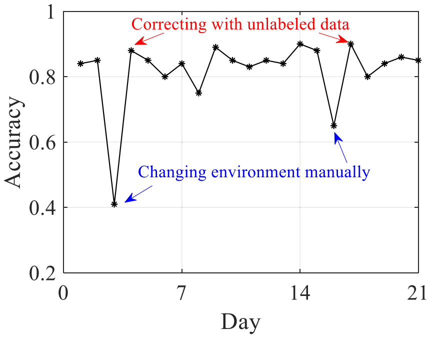


Fig. 21. The classification accuracy of three weeks.

layer number of neural networks can giver better localization results accompanying, it comes with increased training time, as shown in Table X. There is a tradeoff between layer number and training time in application, because the increase of layer will result in a lot of training time but little improvement in performance. Therefore, the second scheme is best in Table X by considering both training time and accuracy.

Since whole transfer network is mainly trained in offline stage after the fingerprint features change, the offline training may not have any effect on real-time performance. To evaluate real-time performance of proposed method, the average online calibrating time and average online prediction time are given in Table XI after 100 tests.

The calibrating time is the cost of PA algorithm in online fine-tuning, the predicting time consists of two parts: 1D-CAE feature extraction and location prediction. As shown in Table XI, the calibrating time and predicting time is considerably low (around 60ms), which is suitable for real-time localization.

F. Long Localization Performance

To evaluate the long-term performance of proposed localization framework, the localization accuracy is tested respectively in 9:00, 13:00 and 17:00 every day for three weeks. The test scenario is selected as E3, and the other basic experiment parameters are same with Section 5.2.

As shown in Fig. 21, the proposed localization framework can provide stable positioning results over a long period of time. Even though severe environmental changes reduce the positioning accuracy, it can still give reliable results by adding newly collected unlabeled fingerprints or a small number of labeled fingerprints.

VI. Conclusion

In this article, a GPI strategy-based transfer learning framework is proposed to solve localization problem in dynamic environment with only unlabeled fingerprints or few labeled fingerprints. In offline stage, 1-CAE is designed to extract deep and related 1D features between different subcarriers; after that, the GPI strategy-based adversarial training is used for the fusion framework of DANN and PA algorithm, which can realize DA between source domain and target domain using only unlabeled data or few labeled data. In online stage, if there are some fluctuations in localization results, a small amount of labeled data can be collected to calibrate the positioning system with a short training time (less than predicting time). A series of experiments testified that the proposed method outperforms traditional positioning methods. Compared with most of the current works, our scheme still has the advantages of accuracy and stability in dynamic environment positioning. Furthermore, the proposed method reduces the labor cost and time consumption in fingerprint collection, which is conducive to the promotion of Wi-Fi positioning system.

The proposed positioning work can realize the feature transfer of fingerprints relying on presetting error threshold, but it is hard to intelligently track environmental changes and identify the types of changes, which is also ignored by most positioning studies. Our future work will focus on solving this problem.

References

[1] K. Lin, M. Chen, J. Deng, M. M. Hassan, and G. Fortino, “Enhanced fingerprinting and trajectory prediction for IoT localization in smart buildings,” *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1294–1307, Jul. 2016.

[2] X. Guo, N. Ansari, F. Hu, Y. Shao, N. R. Elikplim, and L. Li, “A survey on fusion-based indoor positioning,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 566–594, 1st Quart., 2020.

[3] H. Lu, T. Wang, F. Ge, and Y. Shen, “A robust UWB array localization scheme through passive anchor assistance,” *China Commun.*, vol. 18, no. 4, pp. 1–13, Apr. 2021.

[4] D. R.-Y. Phang, W.-K. Lee, N. Matsuhira, and P. Michail, “Enhanced mobile robot localization with lidar and IMU sensor,” in *Proc. IEEE Int. Meeting Future Electron Devices, Kansai (IMFEDK)*, Nov. 2019, pp. 71–72.

[5] H. Wu, S. He, and S. H. G. Chan, “A graphical model approach for efficient geomagnetism-pedometer indoor localization,” in *Proc. IEEE 14th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2017, pp. 371–379.

[6] S. G. Obreja and A. Vulpe, “Evaluation of an indoor localization solution based on Bluetooth low energy beacons,” in *Proc. 13th Int. Conf. Commun. (COMM)*, Jun. 2020, pp. 227–231.

[7] B. Wang, Y. Wang, X. Qiu, and Y. Shen, “BLE localization with polarization sensitive array,” *IEEE Wireless Commun. Lett.*, vol. 10, no. 5, pp. 1014–1017, May 2021.

[8] M. M. del Horno, I. García-Varea, and L. O. Barbosa, “Calibration of Wi-Fi-based indoor tracking systems for android-based smartphones,” *Remote Sens.*, vol. 11, no. 9, p. 1072, May 2019.

[9] X. Tong, H. Li, X. Tian, and X. Wang, “Wi-Fi localization enabling self-calibration,” *IEEE/ACM Trans. Netw.*, vol. 29, no. 2, pp. 904–917, Apr. 2021.

[10] X. Liu *et al.*, “Kalman filter-based data fusion of Wi-Fi RTT and PDR for indoor localization,” *IEEE Sensors J.*, vol. 21, no. 6, pp. 8479–8490, Mar. 2021.

[11] W. Chengqing, L. Chenning, and X. Haowei, “An improved visual indoor navigation method based on fully convolutional neural network,” in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Aug. 2020, pp. 1–5.

[12] Y. Cao, C. Yang, R. Li, A. Knoll, and G. Beltrame, “Accurate position tracking with a single UWB anchor,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2344–2350.

[13] Y. Zhang, X. Tan, and C. Zhao, “UWB/INS integrated pedestrian positioning for robust indoor environments,” *IEEE Sensors J.*, vol. 20, no. 23, pp. 14401–14409, Dec. 2020.

[14] L. Zheng, W. Zhou, W. Tang, X. Zheng, A. Peng, and H. Zheng, “A 3D indoor positioning system based on low-cost MEMS sensors,” *Simul. Model. Pract. Theory*, vol. 65, pp. 45–56, Jun. 2016.

[15] Q. Tian, K. I.-K. Wang, and Z. Salcic, “An INS and UWB fusion approach with adaptive ranging error mitigation for pedestrian tracking,” *IEEE Sensors J.*, vol. 20, no. 8, pp. 4372–4381, Apr. 2020.

[16] G. Guo, R. Chen, F. Ye, X. Peng, Z. Liu, and Y. Pan, “Indoor smartphone localization: A hybrid WiFi RTT-RSS ranging approach,” *IEEE Access*, vol. 7, pp. 176767–176781, 2019.

[17] B. Huang, R. Yang, B. Jia, W. Li, and G. Mao, "A theoretical analysis on sampling size in WiFi fingerprint-based localization," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3599–3608, Apr. 2021.

[18] J. Hu, D. Liu, Z. Yan, and H. Liu, "Experimental analysis on weight K-nearest neighbor indoor fingerprint positioning," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 891–897, Feb. 2019.

[19] H. Chen, B. Wang, Y. Pei, and L. Zhang, "A WiFi indoor localization method based on dilated CNN and support vector regression," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 165–170.

[20] X. Song, Y. Zhou, H. Qi, W. Qiu, and Y. Xue, "DuLoc: Dual-channel convolutional neural network based on channel state information for indoor localization," *IEEE Sensors J.*, vol. 22, no. 9, pp. 8738–8748, May 2022, doi: 10.1109/JSEN.2022.3160700.

[21] W. Liu, H. Chen, Z. Deng, X. Zheng, X. Fu, and Q. Cheng, "LC-DNN: Local connection based deep neural network for indoor localization with CSI," *IEEE Access*, vol. 8, pp. 108720–108730, 2020.

[22] L. Zheng, B.-J. Hu, J. Qiu, and M. Cui, "A deep-learning-based self-calibration time-reversal fingerprinting localization approach on Wi-Fi platform," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7072–7083, Aug. 2020.

[23] R. Zhou, H. Hou, Z. Gong, Z. Chen, K. Tang, and B. Zhou, "Adaptive device-free localization in dynamic environments through adaptive neural networks," *IEEE Sensors J.*, vol. 21, no. 1, pp. 548–559, Jan. 2021.

[24] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.

[25] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 712, pp. 551–585, Dec. 2006.

[26] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[27] W. Zhao, S. Han, R. Q. Hu, W. Meng, and Z. Jia, "Crowdsourcing and multisource fusion-based fingerprint sensing in smartphone localization," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3236–3247, Apr. 2018.

[28] Y. Li, X. Hu, Y. Zhuang, Z. Gao, P. Zhang, and N. El-Sheimy, "Deep reinforcement learning (DRL): Another perspective for unsupervised wireless localization," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6279–6287, Jul. 2020.

[29] F. Dou, J. Lu, T. Xu, C.-H. Huang, and J. Bi, "A bisection reinforcement learning approach to 3-D indoor localization," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6519–6535, Apr. 2021.

[30] C. Chen, Y. Han, Y. Chen, and K. J. R. Liu, "Indoor global positioning system with centimeter accuracy using Wi-Fi [applications corner]," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 128–134, Nov. 2016.

[31] L. Zhao, H. Huang, and X. Li, "An accurate and robust approach of device-free localization with convolutional autoencoder," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5825–5840, Jun. 2019.

[32] Z. Zhang, J. Liu, L. Wang, and G. Guo, "An enhanced smartphone indoor positioning scheme with outlier removal using machine learning," *Remote Sens.*, vol. 13, no. 6, p. 1103, Nov. 2016.

[33] M. Tu Hoang *et al.*, "A CNN-LSTM quantifier for single access point CSI indoor localization," 2020, *arXiv:2005.06394*.

[34] H. Li, X. Zeng, Y. Li, S. Zhou, and J. Wang, "Convolutional neural networks based indoor Wi-Fi localization with a novel kind of CSI images," *China Commun.*, vol. 16, no. 9, pp. 250–260, Sep. 2019.

[35] C. Han, W. Xun, L. Sun, Z. Lin, and J. Guo, "DSCP: Depthwise separable convolution-based passive indoor localization using CSI fingerprint," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–17, Jan. 2021.

[36] G. Cerar, A. Svigelj, M. Mohorcic, C. Fortuna, and T. Javornik, "Improving CSI-based massive MIMO indoor positioning using convolutional neural network," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit (EuCNC/6G Summit)*, Jun. 2021, pp. 276–281.

[37] Z. Shi, L. Wei, and Y. Xu, "CSI-based fingerprinting for indoor localization with multi-scale convolutional neural network," in *Proc. IEEE 3rd Eurasia Conf. IoT, Commun. Eng. (ECICE)*, Oct. 2021, pp. 233–237.

[38] Y. Jing, J. Hao, and P. Li, "Learning spatiotemporal features of CSI for indoor localization with dual-stream 3D convolutional neural networks," *IEEE Access*, vol. 7, pp. 147571–147585, 2019.

[39] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 316–327, Jan. 2020.

[40] Y. Zhang, C. Wu, and Y. Chen, "A low-overhead indoor positioning system using CSI fingerprint based on transfer learning," *IEEE Sensors J.*, vol. 21, no. 16, pp. 18156–18165, Aug. 2021.

[41] H. Li, X. Chen, J. Wang, D. Wu, and X. Liu, "DAFI: WiFi-based device-free indoor localization via domain adaptation," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 4, pp. 1–21, Dec. 2021.

[42] B. Yang, L. Guo, R. Guo, M. Zhao, and T. Zhao, "A novel trilateration algorithm for RSSI-based indoor localization," *IEEE Sensors J.*, vol. 20, no. 14, pp. 8164–8172, Jul. 2020.

[43] 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 *Proc. IEEE Conf. Comput. Commun.*, Vancouver, BC, Canada, Jul. 2021, pp. 1–10.

[44] S. Tiku and S. Pasricha, "Siamese neural encoders for long-term indoor localization with mobile devices," 2021, *arXiv:2112.00654*.

[45] W. Liu *et al.*, "Survey on CSI-based indoor positioning systems and recent advances," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–8.

[46] J. Yan, C. Ma, B. Kang, X. Wu, and H. Liu, "Extreme learning machine and AdaBoost-based localization using CSI and RSSI," *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1906–1910, Jun. 2021.

[47] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 1–32, 2013.

[48] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the Mona Lisa: Spot localization using PHY layer information," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2012, pp. 183–196.

[49] S. Matsushima, N. Shimizu, K. Yoshida, T. Ninomiya, and H. Nakagawa, "Exact passive-aggressive algorithm for multiclass classification using support class," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2010, pp. 303–314.

[50] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, Jan. 2011.

[51] X. Wang, X. Wang, and S. Mao, "ResLoc: Deep residual sharing learning for indoor localization with CSI tensors," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[52] W. Liu, H. Chen, Z. Deng, X. Zheng, X. Fu, and Q. Cheng, "LC-DNN: Local connection based deep neural network for indoor localization with CSI," *IEEE Access*, vol. 8, pp. 108720–108730, 2020.