Semi-Supervised Deep Adversarial Forest for Cross-Environment Localization

Wei Cui, Le Zhang, Bing Li*, Zhenghua Chen, Min Wu, Xiaoli Li and Jiawen Kang

***Abstract*—Extracting channel state information (CSI) from WiFi signals is of proved high-effectiveness in locating human locations in a device-free manner. However, existing localization/positioning systems are mainly trained and deployed in a fixed environment, and thus they are likely to suffer from substantial performance declines when immigrating to new environments. In this paper, we address the fundamental problem of WiFi-based cross-environment indoor localization using a semi-supervised approach, in which we only have access to the annotations of the source environment while the data in the target environments are un-annotated. This problem is of high practical values in enabling a well-trained system to be scalable to new environments without tedious human annotations. To this end, a deep neural forest is introduced which unifies the ensemble learning with the representation learning functionalities from deep neural networks in an end-to-end trainable fashion. On top of that, an adversarial training strategy is further employed to learn environment-invariant feature representations for facilitating more robust localization. Extensive experiments on real-world datasets demonstrate the superiority of the proposed methods over state-of-the-art baselines. Compared with the best-performing baseline, our model excels with an average 12.7% relative improvement on all six evaluation settings.**

***Index Terms*—Device free, Indoor positioning, Semi-supervised Learning, Deep Learning, Adversarial Learning**

I. Introduction

INDOOR localization is an essential technique that enables pervasive applications in many problems, such as locating vehicles in a multi-storey car park and providing precise position of the objectives in search-and-rescue systems. Recent efforts have been devoted to use various off-the-shelf wireless signals (e.g., Bluetooth, Radio Frequency Identification (RFID), IEEE 802.11 (WiFi) etc.) to perform indoor positioning. Among them, WiFi-based indoor positioning has attracted increasing research interests, mostly owing to its "device-free" character and rapid development of WiFi infrastructures in indoor environments.

Machine learning algorithms have been successfully employed for fingerprint-based indoor localization, such as K-nearest neighbor (KNN) and weighted K-nearest neighbor (WKNN), random forest (RF), and so on [1]. Zhou *et al.* [2] proposed a novel positioning solution which utilizes Support Vector Machines (SVM) to establish the nonlinear relationship between CSI fingerprints and target locations in the physical space covered with WiFi signals. Shi *et al.* [3] proposed a CSI-based passive indoor localization system which uses Bayes classifier-based technique combined with multivariate Gaussian distribution to improve the localization accuracy. Sen *et al.* proposed PinLoc [4], a CSI-based localization approach utilizes subcarrier frequency response as the features of a location, and relies on clustering to locate a spot.

Recently, due to the strong expressiveness, deep neural networks are introduced to enhance the indoor localization performance. Some CSI-based indoor fingerprinting systems, such as DeepFi [5] and PhaseFi [6], employ deep autoencoder networks to generate deep-neural fingerprints. These deep learning based systems require CSI amplitude and CSI calibrated phase to feed as input of the deep neural networks. To improve accuracy, Wang *et al.* [7] proposed CiFi, a CSI-based indoor localization system using deep convolutional neural networks (DCNN). CiFi employs phase data of CSI to create the AOA images and feed them into the DCNN for training. Gap *et al.* [8] proposed a deep learning method to learn features from radio images transformed from CSI measurements to estimate the location of a person. Deep auto-encoder and deep neural network are also successfully employed for indoor localization system using hybrid RSS and CSI fingerprints [9]. Li *et al.* [10] propose a deep Siamese convolution neural network to improve the positioning accuracy for CSI fingerprint-based positioning methods.

It has been observed that better accuracy can be obtained by employing deep learning techniques rather than conventional machine learning techniques with handcrafted features. Nevertheless, prior methods perform the positioning only in a *single* environment that is well-controlled and less-changed, while do not consider the positioning in the *cross-environment* scenario. In localization practices, the environment are often subject to dynamically changes, e.g., changing furnishing. The changing environments usually exhibit diverse spatial layouts and thus the objects in different environments are likely to induce distinct multi-path reflections in wireless signals. However, existing deep learning systems typically ignore this fact and work in a black box manner. Hence, they usually perform poorly when being deployed in new environments. Manually collecting and annotating new data in the new environment and re-training the system could solve the problem. However, this procedure could be costly, time-consuming, error-prone, and requires massive human intervention for every new environment.

By observing this, we investigate the problem of "Semi-Supervised Cross-Environment Device-free Localization". In our setting, the positioning system is firstly trained in one

* Corresponding author

This work was supported by the National Natural Science Foundation of China (61903231), and the Agency for Science, Technology and Research (A*STAR) under Career Development Award (Grant No. C210112046).

Wei Cui, Zhenghua Chen, Min Wu, and Xiaoli Li are with the Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore.

Bing Li is with the Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), Singapore. (email: Li_Bing@ihpc.a-star.edu.sg)

Le Zhang is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China.

Jiawen Kang is with the School of Automation, Guangdong University of Technology.

environment which we call *source environment* and further deployed in a new environment which is called *target environment*. We have access to a set of well-annotated data samples in the source environment and un-annotated data samples in the target environment. This is valuable in practice because it enables the existing system to adapt to a new environment with minimum human effort. Motivated by the recent successes of random forest and ensemble deep learning on the different tasks [11], we proposed a robust positioning model using deep neural forest, which unifies the robust generalization ability of ensemble learning with the discriminative representations in an end-to-end framework to address the vulnerability of WiFi signals to environmental dynamics. A adversarial learning procedure is further used to fine-tune the feature extractor in target environment to generate representative features in the target environment for better localization performances. Real-world experimental results demonstrate the superiority of the proposed method for cross-environment indoor localization utilizing CSI measurements.

II. Methodology

Before we step into the detailed information of the proposed method, we firstly introduce the notations used in this study. We assume that we have access to N^1 training samples and their corresponding labels in the source environment $\mathcal{F} \subset (H^s, P^s)$, where $H^s = \{\mathbf{h}_1^s, \mathbf{h}_2^s, \ldots, \mathbf{h}_{N^1}^s\}, H^s \in \mathcal{H}$ and $P = \{\mathbf{p}_1^s, \mathbf{p}_2^s, \ldots, \mathbf{p}_{N^1}^s\}, P^s \in \mathcal{P}$. $\mathbf{h}_i^s$ is a vector of CSI measurements and $\mathbf{p}_i^s = (a, b)$ is the corresponding coordinate of a reference point. We also have access to a set of un-annotated data $H^t = \{\mathbf{h}_1^t, \mathbf{h}_2^t, \ldots, \mathbf{h}_{N^2}^t\}$. Our objective is to predict their locations and we solve this by learning a mapping function.

An overview of the proposed system is shown in Fig. 1, which could be decomposed into a feature extraction network for source domain E_s, a fine-tuned feature extractor for target domain E_t, an adversarial classifier D and a position predictor parameterized by L. Same as traditional fingerprint-based positioning systems, during the offline training stage, the collected fingerprint database in source domain (CSI measurements and corresponding labels) is leveraged to train the feature extractor and location predictor. Afterward, an adversarial learning procedure is employed to learn environment-invariant attributes with E_t. Then, in the online testing stage, upon receiving query fingerprints, the output of the feature extractor E_t is connected with the position predictor L to calculate the localization results. The details of the network will be elaborated in the following section.

A. Localization using dNFs in source domain

For WiFi-based positioning, the vulnerability of WiFi signals to environmental dynamics is a serious challenge. The performance of the localization method will be affected by variations of humidity, light, temperature, occupancy distribution, and human movement in the environment. To solve this issue, we design a robust indoor positioning model that unifies decision trees with the representation learning functionality known from deep convolutional neural networks in an end-to-end trainable fashion.

Given a fingerprint dataset consisting of N_m reference points in the source environment, each fingerprint sample $\mathbf{h}^s$ is a N_d dimension vector of CSI measurements received

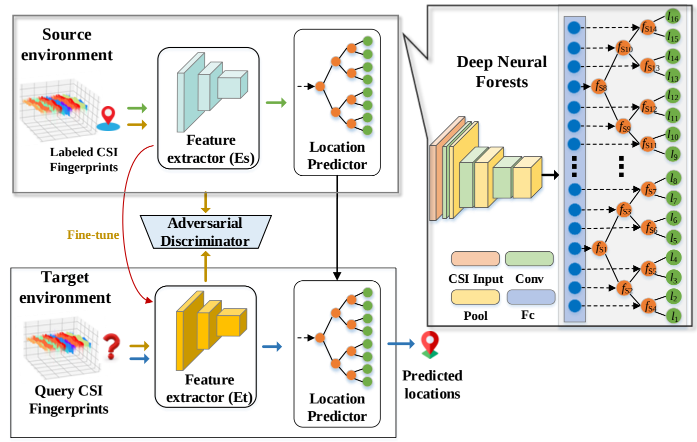

Fig. 1. An illustration of the proposed system. We first pre-train the deep neural forest including the components feature extractor (Es) and location predictor using the fingerprint dataset in source environment. Next, we perform adversarial adaptation by learning a feature extractor (Et) in target environment such that environment discriminator that sees the extracted source and target measurements cannot reliably predict their environment label. During the online testing stage, query CSI fingerprints are mapped with target feature extractor to the shared latent space and regressed by the source location predictor.

from the transmitter during a period of time. In this case, the localization problem could be reformulated as training the mapping function $\mathcal{G} : \mathcal{H} \rightarrow \mathcal{P}$ with input H^s and output $\mathbf{p}^s$ (the corresponding coordinate) in source environment.

In this work, we present a novel deep Neural Forests (dNFs) as our mapping function $\mathcal{G}$ to address the localization regression problem. Consider the proposed deep neural forests consisting of K base decision trees $\mathcal{G} = \{\mathbf{d}_k\}_{k=1}^K$, where each decision tree $\mathbf{d}_k$ is a tree-structured regressor which regresses the location of a sample $\mathbf{h}^s$ by firstly learning high-level representative features from a deep convolutional neural network $\mathcal{N}$, and then the output will be turned into routing decision from the root to leaf nodes, recursively.

1) Feature extractor using CNN: CNN is an essence building blocking for deep learning. Inspired by the organization of visual cortex, CNN structures are analogous to that of the connectivity pattern of neurons in the human brain. It has the demonstrated ability and become the *de-facto* standard in many tasks, such as image recognition, recommender system, and natural language processing. Especially for the image classification task, CNN is a predominant technique, as it is computationally efficient and robust to noise.

Similar to pixel images, the measurements of CSI share conformity. Firstly, there are spatial correlations between different subcarriers in CSI measurements, e.g., subcarriers nearby share similar environmental fluctuations in CSI (i.e., locality). Secondly, the informative pattern is subtle and implicit among the large CSI data. Thirdly, a pattern is translation-invariance among different samples. As such, we employ CNN as the feature extractor $\mathcal{N}$ to extract high-level features from input H^s. Specifically, the architecture of CNN in our model consists of two convolutional layers (Conv), one max-pooling layer (Pool), two Batch Normalization layers (BatchNorm) and two dense layers (Fc).

2) Location predictor using neural forest: In the specific, the forest is an ensemble of decision trees which consist of split (decision) nodes and leaf (prediction) nodes. The split nodes are the tree's internal nodes which are indexed by $\mathcal{S}$, whereas the leaf nodes are the tree's terminal nodes indexed by $\mathcal{L}$. We assume that each split node $S \in \mathcal{S}$ is assigned a splitting function $f_{S_j}(\mathcal{N}(\mathbf{h}^s), \theta_{S_j})$, where θ_{S_j} is the parameter of the splitting function in the split node $S_j \in \mathcal{S}$, $f_{S_j}(\mathcal{N}(\mathbf{h}^s)$ stands for the probability $p_{S_j} \in [0,1]$ of assigning the sample to the left node. When a sample $\mathcal{N}(\mathbf{h}^s)$ reaches a decision node S_j, the split function is responsible to assign the data to the left or right subtree according to the results of $f_{S_j}(\mathcal{N}(\mathbf{h}^s)$. Conventional decision trees are binary and the splitting process is deterministic. On the contrary, we consider a probabilistic routing, i.e., the probability of the left node is Bernoulli Random Variable with mean $f_{S_j}(\mathcal{N}(\mathbf{h}^s)$. When a sample ends at a leaf node $l \in \mathcal{L}$, the prediction of the tree could be obtained by label distribution π_l which is a probability distribution over $\mathcal{N}(H^s)$. In the case of probabilistic routing, the prediction of leaf node is given by the average of the probability at the leaf node, and the final prediction of a tree $\mathbf{d}_k$ could be given by

$$\mathbb{P}_{\mathbf{d}_k}\left[\mathbf{h}^s, \theta, \pi\right] = \sum_{l \in \pounds} \pi_{l_{\mathbf{p}^s}} \mu_l\left(\mathbf{h}^s, \theta\right) \tag{1}$$

where $\pi_{l_{\mathbf{p}^s}}$ indicates the probability of the data with the label $\mathbf{p}^s$, $\mu_l\left(\mathbf{h}^s, \theta\right)$ denotes the routing function which stands for the probability of $\mathbf{h}^s$ reaching at leaf l. The probability of the sample $\mathbf{h}^s$ reaching at leaf l_4 along the black path can be given by

$$\mu_{l_4} = f_{S_1}(\mathcal{N}(\mathbf{h}^s))\left(1 - f_{S_2}(\mathcal{N}(\mathbf{h}^s))\right)\left(1 - f_{S_5}(\mathcal{N}(\mathbf{h}^s))\right) \tag{2}$$

Finally, the prediction of the neural forest $\mathcal{G}$ with trees $\mathbf{d}_1, ..., \mathbf{d}_K$ is given by averaging the output of each tree, i.e.,

$$\mathbb{P}_{\mathcal{G}}\left[\mathbf{p}^s | \mathbf{h}^s\right] = \frac{1}{K} \sum_{i=1}^{K} \mathbb{P}_{\mathbf{d}_k}\left[\mathbf{p}^s | \mathbf{h}^s\right] \tag{3}$$

By the above analysis we can see that training the neural forest mainly requires estimating the parameter of split node θ and the leaf node parametrization π. We use the lowest empirical risk principle to estimate them with regard to the given fingerprint dataset $\mathcal{F}$ in source environment under log-loss looking for the minimizers of the following risk term:

$$R(\theta, \pi, \mathcal{F}) = \frac{1}{\mathcal{F}} \sum_{(\mathbf{h}^s, \mathbf{p}^s) \subset \mathcal{F}} L(\theta, \pi, \mathbf{h}^s, \mathbf{p}^s) \tag{4}$$

where $L(\theta, \pi, \mathbf{h}^s, \mathbf{p}^s))$ is the log-loss term and could be formulated as:

$$L(\theta, \pi, \mathbf{h}^s, \mathbf{p}^s) = -\log(\mathbb{P}_{\mathbf{d}_k}[\mathbf{h}^s, \theta, \pi]) \tag{5}$$

We use the two-step optimization strategy to alternate update the parametrizations θ and π to minimize (4). Please refer to [11] for more details of the two-step optimization strategy.

B. Adversarial Learning for Environment adaptation

As mentioned in Fig.1, our goal is to learn the target feature extractor (E_t) and location predictor (L) that can correctly estimate the location of query CSI measurements during testing phase without annotations. The received CSI measurements may have different structure under different environment conditions even for same person with the same location. Representations learned from each environment individually without adapting discriminative models between them may present significant discrepancy. To this end, we use independent feature extractors for source and target environments and learn only target feature extractor adversarially. To be specific, once the deep neural forest (consisting of feature extractor E_s and location predictor L) is determined in the source environment, we fix the source feature extractor and use adversarial adaptive learning to parametrize the target feature extractor by minimizing the representation distances between the source and target environments, i.e., $\mathcal{N}^s(\mathbf{h}^s)$ and $\mathcal{N}^t(\mathbf{h}^t)$. In this case, then we can directly apply the source location predictor, L, to the target environment without learning a separate target location estimator.

First, an environment discriminator, denoted as D with parameters W^D, is designed to classify which environment the input datum comes from based on high-level features learnt from feature extractor E_s and E_t. Next, since the target feature extractor generally matches source feature extractor in terms of the specific architecture [12], we regularize the target feature extractor with same architecture of source feature extractor. To ensure that the target feature extractor is discriminative when applied in source domain, we fully unty weights between the source and target environment, allowing the target feature extractor learning parameters independently. Besides, in order to achieve effective adaptation and avoid degenerate solutions, we initialize the parameters of target feature extractor with the pre-trained source extractor, and fix the source extractor during adversarial learning. This setting mimics the generative adversarial network (GAN) setting, where the real image distribution remains fixed, and the generated distribution is updated until it is indistinguishable. More details about GAN can be found in [13].

The environment discriminator is optimizated with the loss $\mathcal{L}^{advD}(H^s, H^t, \mathcal{N}^s, \mathcal{N}^t)$, defined as below:

$$\begin{aligned} \min_{D} \mathcal{L}_D^{adv}(W^D, \mathcal{N}^t) = \\ - \sum_{i=1}^{N^1+N^2} \sum_{d \ \in \{0,1\}} \mathbb{I}[y_i = d] \log q_d^D(\mathcal{N}^s(\mathbf{h}^s)) \\ - \sum_{i=1}^{N^1+N^2} \sum_{d \ \in \{0,1\}} \mathbb{I}[y_i = d] \log q_d^D(\mathcal{N}^t(\mathbf{h}^t)) \end{aligned} \tag{6}$$

where $\mathbb{I}$ is an indicator function with $\mathbb{I}(True) = 1$ otherwise 0, q^D corresponding to the softmax of the environment classifier D, that is:

$$q^D = softmax(W^D \cdot \mathcal{N}), \tag{7}$$

and the probability of the d^{th} environment is:

$$q_d^D = \frac{D^{(W^D \cdot \mathcal{N})_d}}{\sum_{d=0}^{1} D^{(W^D \cdot \mathcal{N})_d}} \tag{8}$$

As in [14], an adversarial-like learning objective is further introduced which aims at "maximally confusing" the environment discriminator by computing the cross entropy between the output predicted environment labels, and learning a target feature extractor which generates a distribution matches the source distribution:

$$\begin{aligned} \min_{\mathcal{N}^t} \mathcal{L}_{\mathcal{N}^t}^{adv}(W^D, \mathcal{N}^t) = \\ - \sum_{i=1}^{N^1+N^2} \sum_{d \ \in \{0,1\}} \log q_d^D(\mathcal{N}^t(\mathbf{h}^t)) \end{aligned} \tag{9}$$

The two losses $\mathcal{L}_D^{adv}$ and $\mathcal{L}_{\mathcal{N}^t}^{adv}$ stand in direct opposition to one another. We optimize $\mathcal{L}_D^{adv}$ and $\mathcal{L}_{\mathcal{N}^t}^{adv}$ in an iterative manner. More specifically, we first optimize the W^D in $\mathcal{L}_D^{adv}$ with $\mathcal{N}^t$ being fixed to improve the environment discriminator W^D. Then we optimize $\mathcal{N}^t$ in $\mathcal{L}_{\mathcal{N}^t}^{adv}$ with W^D being frozen, aiming at learning an target feature extractor which could provide domain specific features.

III. Experiments

A. Dataset

We collect the dataset from three typical real-life indoor environments from three different buildings: (a) Environment 1 is an office space covering a $16 \times 20m^2$ area and consisting of standard furniture: tables, chairs, desktops, etc. (b) Environment 2 is an activity space where some fitness equipments crowd in an area of $12 \times 18m^2$. (c) Environment 3 is a classroom space which contains many tables and chairs and covers an area of $14 \times 14m^2$.

In all environments, we use a commercial WiFi router (TP-LINK) with three antennae as a transmitter. For a receiver, we use a Lenovo laptop equipped with an Intel 5300 NICs, running with Ubuntu operating system, and installed with the tool provided in [15]. In our setting, both the transmitter and the receiver are fixed on 1.0-meter-high tripods to enable the propagation of WiFi signals within the environmental area. The whole system works at 2.4 GHz with three antennas of receiver collecting data packets at a transmission rate of 20 Hz. At each time instant, $3\times3\times30$ CSI streams are recorded during the measurements. The data collection details are summarized in Table I. At each point, CSI measurements from 500 packet receptions is collected. Due to the environment dynamics have a large impact on the WiFi signals, the data collection in each scenario is performed in different periods of the duration.

TABLE I
Data collection in different scenarios

#	Building Type	No. of Point	Grid Size (m)	No. of Sample	Duration
1	Office	174	1.5	87K	2 weeks
2	Activity	152	1.5	76K	3 weeks
3	Classroom	62	1.5	31K	1 week

B. Experimental Setup

We coin our method as *TransLoc* and implemented it in Pytorch in Ubuntu 16.04 environment with a 12 GB Titan-X GPU and an Intel i7 CPU (3.4 GHZ). For the *deep neural forest training* in source domain, the epoch number is set to be 300. The batchsize is set to be 32. For the feature extractor, we also use dropout ($p = 0.5$) after each max-pooling layer with Leaky-ReLU ($\alpha = 0.1$) as activate function. The maximum tree depth is set to be 6.

To verify the effectiveness of the proposed approach, we perform extensive experiments in different scenarios and compare them with state-of-the-art semi-supervised learning methods in the literature. In [16], *TCA* shows remarkable performances in cross-domain indoor WiFi localization. In [17], *Deep CORAL* shows superior performances over semi-supervised learning approaches: *CORAL* [18], *DDC* [19], *DAN* [20]. Therefore, we compared our proposed model with *TCA* and *Deep CORAL*. To extensively evaluate the performance of the proposed model, we also compare *TransLoc* with baseline methods, *CNNLoc* [21] and *DeepFuzzy* [22], in which we train the system in one environment and directly test it in the new environment. The parameters of all the approaches are carefully tuned using a validation set from the training data.

For the performance criteria, we adopt the widely used Root Mean Squared Error (RMSE), mean absolute error (MAE) and Standard Deviation (STD).

TABLE II
Execution time of "Environment 1 → Environment 2"

Method	Training time(s)	Testing time(s)
TCA	2225.72	0.05
Deep CORAL	436.54	0.04
CNNLoc	156.59	0.06
DeepFuzzy	161.78	0.07
TransLoc	2140.36	0.14

C. Performance Evaluation

We test the *transferability across environment* of the proposed method with different settings. We treat environment 1 and environment 2 or 3 as the source and target environment respectively and vice versa. Hence in total we have done 6 sets of experiments which will be discussed in the following sections. The experimental results for all 6 sets are summarized in Table III.

For the "Environment 1 $\rightarrow$ Environment 2", our model performed best in terms of RMSE (2.396m) and reached the second-best on MAE (1.479m) and STD (1.095m). Other two transfer learning methods TCA and DeepCORAL achieved an average RMSE around 3m (3.062m, 3.070m), followed by the baseline methods CNNLoc and DeepFuzzy, which achieved RMSE of 4.254m and 3.979m, respectively. Compared to the basic train-once methods CNNLoc and DeepFuzzy, transfer learning based approaches TCA, DeepCORAL and TransLoc can effectively enhance localization accuracy, indicating that the environment-related features lead to performance decline. Generally speaking, it shows that our proposed TransLoc achieves the best localization performance. The similar comparisons are conducted in other settings. Overall, we observe that the proposed method consistently realizes the best localization precision compared with the baseline methods and transfer learning methods in those cases, demonstrating the superiority of our proposed TransLoc. What's more, the transfer learning based methods show better performances than the baseline methods. The reason lies in that naively applying localization methods without learning an environment-invariant representation, could further degenerate the final performance.

Table II shows the empirical execution time. The training time and testing time for our proposed TransLoc are 2140.36s and 0.14s respectively. Please note that the training phase is actually performed offline, the cost of training does not impact real-time positioning. The testing time is the overall cost for all testing samples (76K), the decoding speed is 542.86K/sec, therefore, the proposed TransLoc could well-satisfy the real-world application.

IV. Conclusion

In this paper, we investigate the challenging yet valuable problem of transferring the learned knowledge from a source environment to a target environment for WiFi based device-free localization. In this setting we only have access to

TABLE III
CROSS ENVIRONMENT EVALUATION. WE CONDUCT THE EXPERIMENTS IN THREE DIFFERENT ENVIRONMENTS. FOR EXAMPLE, "ENVIRONMENT 1 → ENVIRONMENT 2" DENOTES ENVIRONMENT 1 AND ENVIRONMENT 2 ARE TREATED AS SOURCE AND TARGET ENVIRONMENT, RESPECTIVELY.

Method	RMSE (m)	MAE (m)	STD (m)	Method	RMSE (m)	MAE (m)	STD (m)
Cross Environment Evaluation Between Environment 1 and Environment 2.							
"Environment 1 → Environment 2"				"Environment 2 → Environment 1"			
TCA	3.062	1.920	1.396	TCA	3.461	2.209	1.803
Deep CORAL	3.070	**1.420**	**1.015**	Deep CORAL	3.730	2.322	1.743
CNNLoc	4.254	2.241	1.654	CNNLoc	5.054	2.861	2.054
DeepFuzzy	3.979	2.049	1.502	DeepFuzzy	4.787	2.834	1.864
TransLoc	**2.396**	1.479	1.095	TransLoc	**2.943**	**1.867**	**1.106**
Cross Environment Evaluation Between Environment 1 and Environment 3.							
"Environment 1 → Environment 3"				"Environment 3 → Environment 1"			
TCA	2.374	1.504	1.246	TCA	3.007	1.934	1.446
Deep CORAL	2.173	2.030	1.460	Deep CORAL	2.873	1.961	**1.043**
CNNLoc	3.853	1.821	1.502	CNNLoc	3.952	**1.821**	1.504
DeepFuzzy	3.490	1.479	1.313	DeepFuzzy	3.708	2.356	1.479
TransLoc	**1.934**	**1.453**	**0.912**	TransLoc	**2.701**	1.869	1.106
Cross Environment Evaluation Between Environment 2 and Environment 3.							
"Environment 2 → Environment 3"				"Environment 3 → Environment 2"			
Method	RMSE (m)	MAE (m)	STD (m)	Method	RMSE (m)	MAE (m)	STD (m)
TCA	3.164	1.967	1.482	TCA	3.555	2.278	1.543
Deep CORAL	3.041	1.962	1.177	Deep CORAL	3.223	2.074	1.234
CNNLoc	4.324	2.456	1.984	CNNLoc	4.256	2.412	2.054
DeepFuzzy	4.011	2.374	1.531	DeepFuzzy	3.928	2.534	1.186
TransLoc	**2.851**	**1.858**	**1.141**	TransLoc	**2.700**	**1.613**	**1.022**

unlabelled data samples in the target environment. We propose a solution for this problem which consists of three stages. In the first stage, the system learns a localization model using deep neural forests with the labelled dataset in the source environment, and is further segregated into source feature extractor and location predictor. This further facilitate the second stage where an adversarial learning strategy is utilized to train a adaptive target feature extractor for system refinement.. After this, we directly apply the source location predictor and the target feature extractor for location estimation in target environment. Extensive experiments are carried out in real-life scenarios to show the effectiveness of the proposed method. In the future, we plan to design more advanced strategies and networks to use the unlabeled data more efficiently and reduce the ratio of labeled data for cross-environment localization.

REFERENCES

[1] W. Liu, Q. Cheng, Z. Deng, H. Chen, X. Fu, X. Zheng, S. Zheng, C. Chen, and S. Wang, "Survey on csi-based indoor positioning systems and recent advances," in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2019, pp. 1–8.

[2] R. Zhou, J. Chen, X. Lu, and J. Wu, "Csi fingerprinting with svm regression to achieve device-free passive localization," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2017, pp. 1–9.

[3] S. Shi, S. Sigg, L. Chen, and Y. Ji, "Accurate location tracking from csi-based passive device-free probabilistic fingerprinting," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5217–5230, 2018.

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

[5] X. Wang, L. Gao, S. Mao, and S. Pandey, "Csi-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2016.

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

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

[8] Q. Gao, J. Wang, X. Ma, X. Feng, and H. Wang, "Csi-based device-free wireless localization and activity recognition using radio image features," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 346–10 356, 2017.

[9] C. Zhou, J. Liu, M. Sheng, Y. Zheng, and J. Li, "Exploiting fingerprint correlation for fingerprint-based indoor localization: A deep learning based approach," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5762–5774, 2021.

[10] Q. Li, X. Liao, M. Liu, and S. Valaee, "Indoor localization based on csi fingerprint by siamese convolution neural network," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12 168–12 173, 2021.

[11] P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulo, "Deep neural decision forests," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1467–1475.

[12] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[14] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," 2015.

[15] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11 n traces with channel state information," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 53–53, 2011.

[16] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE transactions on neural networks*, vol. 22, no. 2, pp. 199–210, 2010.

[17] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 443–450.

[18] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[19] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[20] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.

[21] X. Feng, K. A. Nguyen, and Z. Luo, "A survey of deep learning approaches for wifi-based indoor positioning," *Journal of Information and Telecommunication*, pp. 1–54, 2021.

[22] L. Zhang, Z. Chen, W. Cui, B. Li, C. Chen, Z. Cao, and K. Gao, "Wifi-based indoor robot positioning using deep fuzzy forests," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10 773–10 781, 2020.