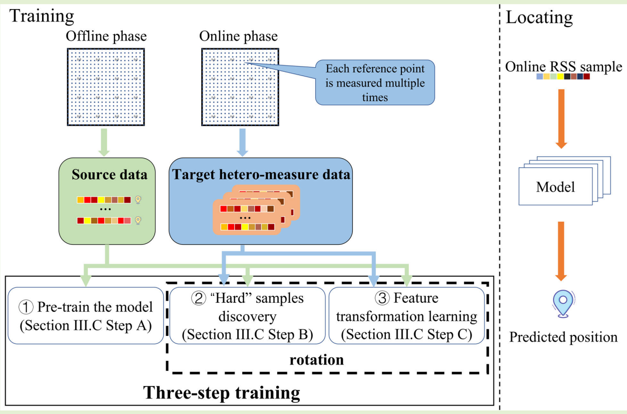
TRAIL: A Three-Step Robust Adversarial Indoor Localization Framework

Yin Yang, Xiansheng Guo, *Senior Member, IEEE*, Cheng Chen, Gordon Owusu Boateng, *Member, IEEE*, Haonan Si, *Graduate Student Member, IEEE*, Bocheng Qian, and Linfu Duan

***Abstract*—Indoor localization utilizing received signal strength (RSS) fingerprint has garnered significant attention over the past decade because it is readily captured from the MAC layer of ubiquitous hardware devices. However, the localization accuracy of RSS fingerprint-based methods is notably influenced by two primary factors: 1) disparities between offline and online data distributions induced by dynamic environmental changes and device heterogeneity and 2) inconsistencies among hetero-measure samples (different RSS samples collected at the same reference point (RP) during the online stage) stemming from unknown noise and interference. To address these issues, we propose a three-step robust adversarial indoor localization (TRAIL) framework. The model is pretrained in the first step (Step A), and an adversarial game is played between a regressor and a feature extractor within the model in the second step (Step B) and third step (Step C). Specifically, Step B trains the regressor to discover more "hard" samples, i.e., hetero-measure samples with notable positioning differences, and Step C trains the feature extractor to learn a suitable transformation that eliminates the disparities between offline and online data distributions and the "hard" samples. To harmonize the contributions of the two factors in model training, we integrate the multiple gradient descent algorithm (MGDA). Experimental results on both actual and simulated datasets demonstrate that TRAIL outperforms state-of-the-art methods and exhibits robustness in low signal-to-noise ratio (SNR) environments.**


Training
Offline phase
Online phase
Each reference point is measured multiple times
Source data
Target hetero-measure data
① Pre-train the model (Section III.C Step A)
② "Hard" samples discovery (Section III.C Step B)
③ Feature transformation learning (Section III.C Step C)
rotation
Three-step training
Locating
Online RSS sample
Model
Predicted position

***Index Terms*— Adversarial learning, indoor localization, received signal strength (RSS), transfer learning.**

NOMENCLATURE

$\mathcal{D}_S$, $\mathcal{D}_T$	Source/target domain.
n_S, n_T	Number of samples of source domain/RPs of target domain.
$\mathbf{X}_S$, $\mathbf{X}_T$	Source feature matrix/target feature tensor.
$\mathbf{X}_T(j, :, :)$	Hetero-measure matrix of the jth RP in target domain.
$\mathbf{X}_S^f$, $\mathbf{X}_T^f$	Extracted feature of source/target samples.
$\mathbf{Y}_S$, $\hat{\mathbf{Y}}_S$, $\hat{\mathbf{Y}}_T$	True positions of source samples, predict positions of source/target samples.
$\boldsymbol{y}_S^i$, $\hat{\boldsymbol{y}}_S^i$, $\hat{\boldsymbol{y}}_T^i$	True position of the ith source sample, predict position of the ith source/target sample.
θ_e, θ_r	Parameters of feature extractor/regressor.
L_{hm}, L_{mmd}, L_S	Hetero-measure/multi-kernel maximum mean discrepancy/localization loss function.
L_A, L_B, L_C	Loss function of Step A/B/C.
α, β, μ, ν, ω	Weights of different losses.

I. INTRODUCTION

WITH the rapid development of the Internet of Things (IoT) paradigm, the demand for precise user and IoT device localization has surged. Location-based services

Manuscript received 10 December 2023; accepted 3 January 2024. Date of publication 22 February 2024; date of current version 2 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62171086 and in part by the Municipal Government of Quzhou under Grant 2022D001. The associate editor coordinating the review of this article and approving it for publication was Dr. Hassen Fourati. *(Corresponding author: Xiansheng Guo.)*

Yin Yang, Gordon Owusu Boateng, Haonan Si, Bocheng Qian, and Linfu Duan are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: yinyang@std.uestc.edu.cn; boatenggordon48@gmail.com; sihaonan@std.uestc.edu.cn; bochengqian@std.uestc.edu.cn).

Xiansheng Guo is with the Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, Zhejiang 324000, China, and also with the Department of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: xsguo@uestc.edu.cn).

Cheng Chen is with the Department of Automation, Tsinghua University, Beijing 100084, China, and also with Chengdu Yibo Information Technology Company Ltd., Chengdu, Sichuan 610093, China (e-mail: chenc23@mails.tsinghua.edu.cn).

Digital Object Identifier 10.1109/JSEN.2024.3352669

TABLE I
HETERO-MEASURE SAMPLES OF THE SAME RP

BS1	BS2	BS3	BS4	BS5	BS6	BS7	BS8	BS9	BS10
-93(dBm)	-65(dBm)	-67(dBm)	-85(dBm)	-88(dBm)	-87(dBmm)	-67(dBm)	-86(dBm)	-66(dBm)	-88(dBm)
-88(dBm)	-64(dBm)	-68(dBm)	-88(dBm)	-86(dBm)	-90(dBm)	-68(dBm)	-88(dBm)	-66(dBm)	-86(dBm)
-90(dBm)	-63(dBm)	-66(dBm)	-87(dBm)	-88(dBm)	-91(dBm)	-67(dBm)	-85(dBm)	-64(dBm)	-85(dBm)
-100(dBm)	-64(dBm)	-68(dBm)	-88(dBm)	-100(dBm)	-88(dBm)	-66(dBm)	-88(dBm)	-63(dBm)	-86(dBm)
-100(dBm)	-65(dBm)	-67(dBm)	-85(dBm)	-87(dBm)	-89(dBm)	-66(dBm)	-89(dBm)	-65(dBm)	-87(dBm)

are becoming increasingly integral to various applications, necessitating accurate indoor positioning methods [1]. Existing indoor positioning systems utilize wireless technologies such as Bluetooth, Ultrasonic, WiFi, and radio frequency identification (RFID), and employ one of three fundamental positioning techniques: geometry-based methods [2], pedestrian dead reckoning (PDR) [3], and fingerprint-based methods [4]. Among these techniques, WiFi fingerprint-based localization strategies have garnered significant attention due to the widespread deployment of WiFi infrastructures and the fact that they do not require any initial location information.

The fingerprint-based localization method consists of two stages: offline and online. During the offline stage, the indoor environment is partitioned into multiple reference points (RPs), and the signal characteristics (such as received signal strength (RSS) [5] and channel state information (CSI) [6]) at each RP are recorded to establish a database. During the online phase, various algorithms (e.g., maximum likelihood [7], expectation-maximization [8], and random forest [9]) are utilized to match the signal characteristics measured by the user with the database to estimate the location.

Due to various factors, such as heterogeneous hardware and dynamic changes in the indoor environment, there is typically a significant deviation between the distribution of online samples and the offline database. This deviation can lead to prediction errors when applying the model trained with offline data directly to the online samples. Therefore, it is imperative to frequently collect a large amount of data to rebuild the offline database, which incurs high costs. Most current fingerprint-based indoor localization algorithms, such as crowdsourcing-based methods [10] and transfer learning-based methods [11], aim to address this issue.

However, even when taking multiple measurements of the same RP at the same time with the same hardware, the resulting RSS samples are not necessarily identical due to various factors, including but not limited to multipath interference, channel noise, human shadow effect, and their combined effect. In the rest of this article, we refer to these samples as *“hetero-measure samples.”* To illustrate the influence of these unidentified interferences, we intercept the RSS data obtained by taking measurements of ten base stations (BSs) at the same RP five times in the UJI dataset [12], as shown in Table I. If a BS is not detected in the sample, the corresponding RSS value is -100 dBm. From Table I, it is clear that these interferences not only result in varying RSS measurements from the same BS but also lose the signal of some BSs, leading to inaccurate positioning results.

To effectively exploit the information between hetero-measure samples, we propose a three-step robust adversarial indoor localization (TRAIL) framework. In this framework, we find appropriate cross-domain mappings of source and target domain data, such that a feature space containing common knowledge and accommodating unknown disturbances can be constructed. To achieve this, we first pretrain the model to learn source knowledge in Step A. Then, we train the regressor to discover more “*hard samples*” by maximizing the discrepancies among localization results of hetero-measure samples in Step B. Finally, in Step C, we train the feature extractor to learn the transformation by optimizing the multiple kernel maximum mean discrepancy (MKMMD) function and the discrepancies among localization results of hetero-measure samples simultaneously. In this new feature space, we minimize distribution differences between two domains and place the data to “*soft*” points in the regression hyperplane as much as possible. Here, a “*soft*” point means its derivative is nearly zero in all directions. The key novelty of TRAIL lies in incorporating the reduction of domain distribution discrepancy and the reduction of hetero-measure data differences, eliminating distribution discrepancy and the combined effects of all unknown interferences in the online stage. Additionally, we use the multiple gradient descent algorithm (MGDA) to balance the adversarial training and domain alignment, achieving more accurate localization performance.

In summary, the main contributions of this work are as follows.

1) We design a novel TRAIL framework, named TRAIL, to solve the domain distribution discrepancy problem by optimizing the MKMMD function and enhance the inclusiveness of the model for unknown disturbances by employing maximum-minimum adversarial training to learn the associations and differences between hetero-measure samples. Consequently, the framework can extract device-independent, anti-dynamic features and reduce the influence of unknown interferences to achieve robust localization.
2) Specifically, Step A pretrains the entire model using source domain data to learn the intrinsic relationship between RSS samples and their corresponding physical locations; Step B discovers “*hard*” samples that are heavily affected by unknown interference by training the regressor through maximizing localization differences between hetero-measure data in the target domain; and Step C minimizes the MKMMD loss function between the samples in the source domain and the target domain

and the localization differences of the hetero-measure samples in the target domain to train the feature extractor, so as to find a suitable transformation for precise localization.

3) TRAIL introduces a novel approach to address the challenge of unknown interferences in RSS fingerprint-based indoor localization. Unlike traditional methods that individually address specific interferences, we use an adversarial method to enhance the model's robustness to the collective impact of various interferences. Compared with existing techniques, this innovative strategy yields more precise and dependable localization outcomes.
4) We leverage MGDA to balance the model's capabilities in handling dynamic environmental changes and coping with unknown interferences. This strategy eliminates the need for specific hyperparameters, contributing to the model's robustness and achieving satisfactory results.

The remainder of this article is structured as follows. Section II discusses the related work. Section III presents our proposed framework. In Section IV, extensive experiments are conducted on actual and simulated datasets, and results are provided. Finally, Section V concludes this article.

II. Related Work

We review related works from three perspectives: transfer learning, adversarial learning, and fingerprint-based localization with hetero-measure samples.

A. Transfer Learning

Transfer learning is a machine learning method that transfers knowledge from a source domain to a target domain, resulting in improved performance and reduced training costs. According to [13], most transfer learning algorithms can be primarily categorized into instance-based or feature-based.

Feature-based transfer learning algorithms seek to find a feature transformation that can project the data of two domains into a new feature space with a slight distribution difference. To achieve this, distance metrics such as maximum mean discrepancy (MMD) [14], correlation alignment (CORAL) [15], and Wasserstein distance [16] are commonly used. Pan et al. [17] used kernel methods to minimize the MMD function between domains. Sun et al. [18] proposed a dimension reduction algorithm by manifold learning, which also narrows the distance between the marginal probability distributions of the two domains. Methods like DAN [19] and DeepCoral [15]-integrated MMD or CORAL function with deep learning to take advantage of neural networks' strong fitting capability. However, these methods only align the marginal or conditional probability distribution of the two domains, ignoring the differences between samples within the same class.

Instance-based transfer learning methods weight the source domain data to emphasize samples similar to the target domain data. However, the selection of weights and the measure of sample similarity significantly impact these strategies. In practice, instance-based transfer learning methods are often combined with other methods to achieve better performance. For instance, Transloc [20] combined source domain refinement and homogeneous feature space construction to solve the problem of heterogeneous feature dimensions caused by BS changes in long-term localization tasks and obtained acceptable results. Li et al. [21] reduced data distribution divergences between different domains through multilevel knowledge transfer, including sample, feature, and model levels. Guo et al. [22] calculated the similarity between source and target domain data using the Euclidean distance and maximized the likelihood function to obtain the most probable label estimation.

B. Adversarial Learning

Adversarial learning trains the model to defeat a hypothetical adversary attempting to perturb the input, thereby enhancing the model's robustness and generalization ability. Various methods have been investigated to generate adversarial samples for input perturbation [23], [24]. Except for adversarial learning based on generating adversarial samples, Ganin et al. [25] were the first to propose combining adversarial ideas and domain adaptation, laying the foundation for adversarial transfer learning. Based on this concept, a series of studies have been conducted [26], [27]. Besides, Saito et al. [28] identified hard target samples that are difficult to classify using two distinct classifiers and updated the feature extractor to handle them.

Adversarial learning has found extensive applications in indoor fingerprint localization. Liu and Wang [29] introduced a novel tensor GAN that leveraged a three-person adversarial game involving a generator, a regressor, and a discriminator. Li et al. [30] employed the DANN [25] concept to tackle the knowledge transfer issue for indoor localization and developed a method to generate 2-D RSS fingerprints. In [31], a domain adversarial graph convolutional network was proposed that combined adversarial ideas with graph convolutional networks.

C. Fingerprint-Based Localization With Hetero-Measure Samples

RSS at a specific location varies for various reasons, including multipath effect, human body shadow effect, channel noise, etc. There are already many studies that analyze the above-mentioned interferences separately. For instance, to address or mitigate the impact of the multipath effect, Zhao et al. [32] utilized the discrete Fourier transform to partition the signal into distinct frequency bins and built an RSS model that considered all the dynamics changes. Fang et al. [33] translated dynamic multipath effects into random additive noise in the logarithmic spectrum and provided a way to reduce it. Fang and Chen [34] utilized a three-edged constraint to prevent the traditional multichannel localization algorithm from falling into local optima. To eliminate the impact of the human body shadow effect, Husen and Lee [35] collected the face orientation of a measurer while collecting fingerprints. Bi et al. [36] used a fuzzy C-means (FCM) clustering algorithm to create an orientation fingerprint database (OFPD) in the offline phase. The appropriate OFPD was selected according to the user's orientation, and further matching operations were carried out in the online stage.

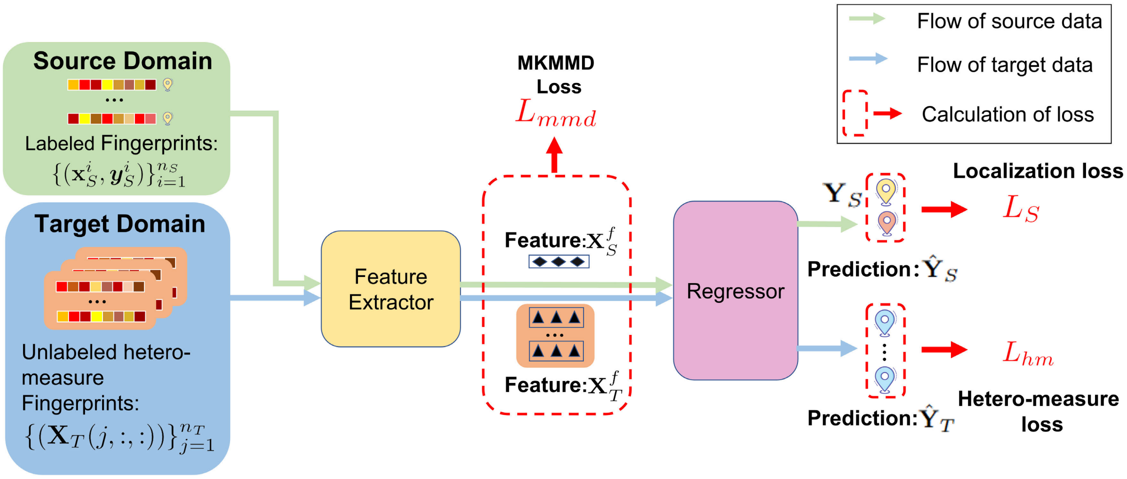


Fig. 1. Model structure diagram. The model includes a feature extractor and a regressor. The green and blue lines represent the flow of source and target data, respectively. The red dotted boxes represent the data needed to calculate different losses.

However, none of these works jointly considers these interferences on RSS. In fact, due to the complexity of indoor environments, most of these interferences act together, so considering one interference alone is unsuitable for actual tasks. In addition to these methods that reduce a particular type of interference individually, other positioning methods either average the hetero-measure samples or treat them as individual fingerprints. As a result, the correlation and distinctive information among the hetero-measure samples are underutilized.

Unlike existing approaches that either consider only the distribution differences between domains or mitigate only one type of interference during the online phase, TRAIL utilizes a small amount of unlabeled hetero-measure data during the online phase to eliminate differences in the distribution between domains and discrepancies in hetero-measure samples without extra information simultaneously. The underlying mechanism is to construct a suitable transformation that can help us perform knowledge transfer efficiently and enhance the tolerance of the model to unknown interference. Experimental results demonstrate the superiority of TRAIL even in low signal-to-noise ratio (SNR) environments.

III. System Framework

In this section, we present the framework of TRAIL. Section III-A briefly introduces the problem setting and our model. In Section III-B, we discuss handling hetero-measure samples. Then, we describe our three-step adversarial training approach in Section III-C. Finally, we explain our employment of MGDA in Section III-D.

A. Overview

Let $\mathcal{D}_S = \{\mathbf{X}_S, \mathbf{Y}_S\} = \{(\mathbf{x}_S^i, \boldsymbol{y}_S^i)\}_{i=1}^{n_S}$ be the labeled source domain collected during the offline phase, where n_S is the number of samples, $\mathbf{x}_S^i$ and $\boldsymbol{y}_S^i$ are the RSS vector and the 2-D coordinates of the ith sample. Each sample consists of RSS values of C BSs. Among them

$$\mathbf{x}_S^i = \left[x_S^{i,1}, x_S^{i,2}, \ldots, x_S^{i,C}\right]^T \tag{1}$$

where $x_S^{i,c}$ denotes the RSS value of the cth BS in the ith sample. Because we need to learn the unknown interferences of the target domain rather than the source domain, we just treat the source domain data as independent fingerprints rather than hetero-measure data.

Similarly, let $\mathcal{D}_T = \{\mathbf{X}_T\} = \{(\mathbf{X}_T(j,:,:))\}_{j=1}^{n_T}$ be the unlabeled target domain collected during the online phase, where n_T is the number of measured RPs, which is not necessarily equal to the total number of RPs, and $\mathbf{X}_T(j,:,:)$ is the hetero-measure matrix at the jth RP. Notably, there are at least two hetero-measure samples at each RP as follows:

$$\begin{aligned} \mathbf{X}_T(j,:,:) &= \left[\mathbf{x}_T^{j,1}, \mathbf{x}_T^{j,2}, \ldots, \mathbf{x}_T^{j,k}\right]^T, k \geq 2, \\ \mathbf{x}_T^{j,k} &= \left[x_T^{j,k,1}, x_T^{j,k,2}, \ldots, x_T^{j,k,C}\right]^T \end{aligned} \tag{2}$$

where $\mathbf{x}_T^{j,k}$ denotes the kth measurement at the jth RP.

As illustrated in Fig. 1, the model proposed in this article primarily comprises a feature extractor and a regressor. The feature extractor is intended to learn a feature transformation that transforms source domain and target domain data into anti-dynamic and anti-interference features $\mathbf{X}_S^f$ and $\mathbf{X}_T^f$, respectively. The regressor uses the extracted features to predict the position. Let θ_e and θ_r be the set of parameters of the feature extractor and regressor, respectively. Let F and R represent the feature extractor and regressor function. Given the source samples $\mathbf{X}_S$, we can obtain predicted positions $\hat{\mathbf{Y}}_S$ as follows:

$$\begin{aligned} \mathbf{X}_S^f &= F(\mathbf{X}_S; \theta_e) \\ \hat{\mathbf{Y}}_S &= R\left(\mathbf{X}_S^f; \theta_r\right). \end{aligned} \tag{3}$$

B. Hetero-Measure Samples Utilization

When performing regression tasks, it is essential to constrain the absolute value of the derivative of the regression curve at the sample point to avoid unstable outputs. If the absolute value of the derivative is too large, even tiny disturbances to the samples can result in significant errors in the output prediction.

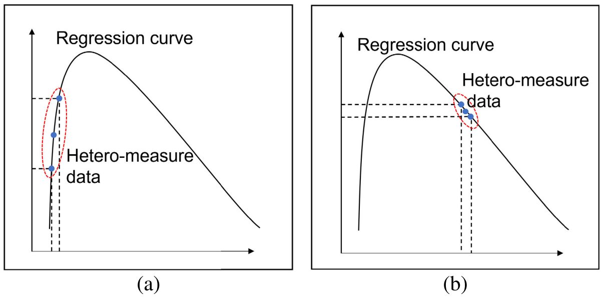


Fig. 2. Different hetero-measure samples in regression curve. (a) Absolute value of the derivative of the regression curve is more significant at the "*hard*" points, and the slight error of the samples leads to the vast difference in the regression value. (b) Absolute value of the derivative of the regression curve is smaller at the "*soft*" points, so more stable regression values can be obtained.

These problematic samples are referred to as "*hard*" samples, as shown in Fig. 2(a). Conversely, we prefer the "*soft*" samples to obtain more stable regression values, as shown in Fig. 2(b). Therefore, we need to make all the samples "*softer*." To achieve this, we can learn a transformation that maps samples into the "*soft*" points in the regression curve.

In the context of our fingerprint localization, the hetero-measure samples can be assumed to be in the same neighborhood. If the regression positions of hetero-measure samples at the same RP are close enough, it can be considered that the hetero-measure samples are "*soft*." To measure the discrepancies among the predicted positions of hetero-samples, we utilize mean squared error (MSE) loss (referred to as hetero-measure loss L_{hm} in the remainder of this article) rather than variance or covariance. We prefer the MSE loss because the variance solely characterizes differences in the predicted positions along a singular dimension within the 2-D coordinates, while the covariance only represents the correlation between two dimensions. However, we are concerned with the overall dispersion between the regression locations of hetero-measure samples, so the sum of Euclidean distances may be more appropriate. To minimize the hetero-measure loss, we can train the feature extractor to learn the transformation and thus improve robustness to the unidentified interference. It is worth noting that we do not need to find out which samples are "*hard*"; what we do is to minimize the hetero-measure loss of the whole target domain to learn that transformation.

C. Three-Step Adversarial Training

To further improve the training effect, we adopt a three-step training process, as shown in Fig. 3.

Step A: In this step, we pretrain the entire model with source data to minimize the localization loss L_S. Our optimization goal is to minimize the loss function L_A in Step A, which is expressed as follows:

$$\begin{aligned}\min_{\theta_e,\theta_r} L_A &= \min_{\theta_e,\theta_r} L_S\\ &= \min_{\theta_e,\theta_r}\left(\frac{1}{n_S}\sum_{i=1}^{n_S}\|\hat{\boldsymbol{y}}_S^i-\boldsymbol{y}_S^i\|_2^2\right)\end{aligned}\tag{4}$$

where θ_e denotes the parameters of the feature extractor, θ_r represents the parameters of the regressor, and $\hat{\boldsymbol{y}}_S^i$ represents the predicted position of the ith source sample.

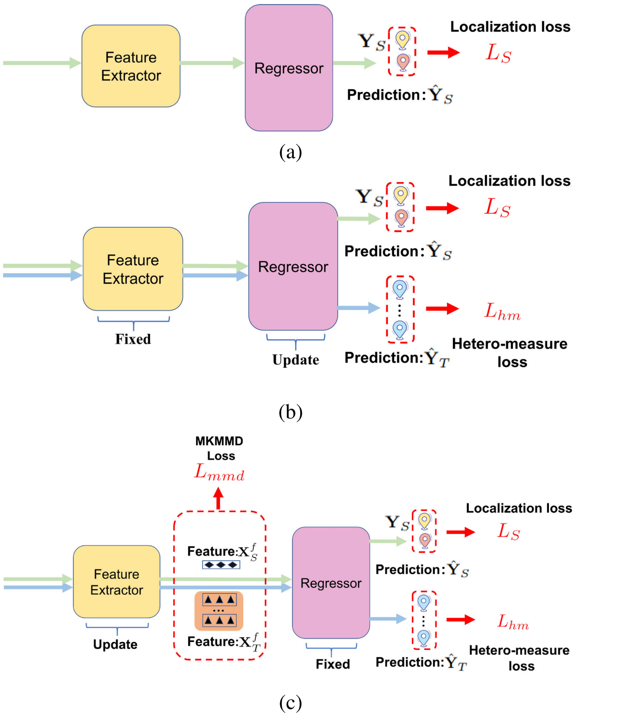


Fig. 3. Three-step adversarial training. (a) Step A. (b) Step B. (c) Step C.

Step B: In this step, we aim to train the regressor to detect "*hard*" samples as much as possible. To achieve this, we freeze the feature extractor and train the regressor by maximizing the hetero-measure loss, i.e., maximize the divergences among the predicted positions of hetero-measure samples. The hetero-measure loss is expressed as follows:

$$\begin{aligned}&\max_{\theta_r} L_{\text{hm}}\\ &\quad=\max_{\theta_r}\frac{1}{n_T\times C_k^2}\sum_{j=1}^{n_T}\sum_{m=1}^{k}\sum_{n=m+1}^{k}\|\hat{\boldsymbol{y}}_T^{j,m}-\hat{\boldsymbol{y}}_T^{j,n}\|_2^2\\ &\quad=-\min_{\theta_r}\frac{1}{n_T\times C_k^2}\sum_{j=1}^{n_T}\sum_{m=1}^{k}\sum_{n=m+1}^{k}\|\hat{\boldsymbol{y}}_T^{j,m}-\hat{\boldsymbol{y}}_T^{j,n}\|_2^2\end{aligned}\tag{5}$$

where $\hat{y}_T^{j,m}$ represents the predicted position of the mth measurement of the jth RP and $C_k^2=(2!/k!(k-2)!)$.

Meanwhile, we optimize L_S so the regressor can retain the knowledge learned from the source domain. This means the regression hyperplane is still suitable for positioning tasks rather than just focusing on finding as many "*hard*" samples as possible. The total loss function L_B of Step B is as follows:

$$\begin{aligned}&\min_{\theta_r} L_B\\ &\quad=\min_{\theta_r}(\alpha L_S-\beta L_{\text{hm}})\\ &\quad=\min_{\theta_r}\Bigg(\alpha\frac{1}{n_S}\sum_{i=1}^{n_S}\|\hat{\boldsymbol{y}}_S^i-\boldsymbol{y}_S^i\|_2^2\\ &\qquad-\beta\frac{1}{n_T\times C_k^2}\sum_{j=1}^{n_T}\sum_{m=1}^{k}\sum_{n=m+1}^{k}\left(\|\hat{\boldsymbol{y}}_T^{j,m}-\hat{\boldsymbol{y}}_T^{j,n}\|_2^2\right)\Bigg)\end{aligned}\tag{6}$$

where α and β are the adaptive weights learned by the following MGDA algorithm for L_S and L_{hm}, respectively.

Step C: In this step, the regressor is fixed. We train the feature extractor to find the suitable transformation that aligns the distribution of the source and target domains (achieved by minimizing the MKMMD loss function L_{mmd} between offline and online data) and map samples to "*soft*" points (achieved by minimizing the hetero-measure loss function L_{hm} of the online data). Besides, to ensure that the extracted features are not only "*softer*" and close to the source data but also effective, i.e., divisible and conducive to the positioning task, the offline localization loss function L_S is added in L_C as follows:

$$\begin{aligned}
&\min_{\theta_e} L_C \\
&= \min_{\theta_e} \left(\mu L_S + \nu L_{\text{hm}} + \omega L_{\text{mmd}} \left(\mathbf{X}_S^f, \mathbf{X}_T^f\right)\right) \\
&= \min_{\theta_e} \Bigg(\mu \sum_{i=1}^{n_S} \| \hat{\boldsymbol{y}}_S^i - \boldsymbol{y}_S^i \|_2^2 \\
&\qquad + \nu \frac{1}{n_T \times C_k^2} \sum_{j=1}^{n_T} \sum_{m=1}^{k} \sum_{n=m+1}^{k} \| \hat{\boldsymbol{y}}_T^{j,m} - \hat{\boldsymbol{y}}_T^{j,n} \|_2^2 \\
&\qquad + \omega \| \frac{1}{n_S} \sum_{i=1}^{n_S} \Phi\left(\mathbf{x}_S^{f,i}\right) - \frac{1}{n_T \times k} \sum_{j=1}^{n_T \times k} \Phi\left(\mathbf{x}_T^{f,j}\right) \|_2^2 \Bigg) \qquad (7)
\end{aligned}$$

where μ, ν, and ω are the adaptive weights learned by the following MGDA algorithm for L_S, L_{hm}, and L_{mmd}, respectively, L_{mmd} represents the MKMMD loss function, $\mathbf{X}_S^f, \mathbf{X}_T^f$ are extracted features of source and target samples, and $\Phi(\cdot)$ is the multikernel map of MKMMD

$$\begin{aligned}
\mathbf{K}\left(\mathbf{x}_S^{f,i}, \mathbf{x}_T^{f,j}\right) &= \Phi\left(\mathbf{x}_S^{f,i}\right) \cdot \Phi\left(\mathbf{x}_T^{f,j}\right) \\
\mathbf{K} &\triangleq \left\{ \sum_{u=1}^{p} \lambda_U . K_{\cdot u}; \sum_{u=1}^{p} \lambda_u = 1, \lambda_u \geq 0 \right\} \qquad (8)
\end{aligned}$$

where λ_u is the weight and k_u denotes the uth different kernel function.

D. Multiple Gradient Descent Algorithm

In the three-step training process described above, the optimization objectives of both Step B and Step C consist of multiple loss functions. The weights of these losses have an enormous impact on the localization precision of the model. To handle this, our model employs the MGDA to alter the weights dynamically, thus eliminating the need for hyperparameters. To balance the computational complexity and the retention of the source domain knowledge, we set L_B and L_C as follows:

$$\begin{aligned}
L_B &= (1-\beta) L_S - \beta L_{\text{hm}} \\
L_C &= L_S + \nu L_{\text{hm}} + (1-\nu) L_{\text{mmd}}. \qquad (9)
\end{aligned}$$

Then, we dynamically compute β and ν by the MGDA method.

For the two-loss function weights optimization problem [37]

$$\min_{\theta_{\text{sh}}, \theta_1, \theta_2} (\alpha L_1 + (1-\alpha) L_2) \qquad (10)$$

where θ_{sh} denotes the shared parameters for both loss functions, and θ_1 and θ_2 represent the exclusive parameters of L_1 and L_2, respectively.

The optimal $\hat{\alpha}$ of the above problem is given by

$$\begin{aligned}
\hat{\alpha} &= \left[\frac{\left(\nabla_{\theta_{\text{sh}}} L_2 (\theta_{\text{sh}}; \theta_2) - \nabla_{\theta_{\text{sh}}} L_1 (\theta_{\text{sh}}; \theta_1)\right)^{\top} \nabla_{\theta_{\text{sh}}} L_2 (\theta_{\text{sh}}; \theta_2)}{\| \nabla_{\theta_{\text{sh}}} L_1 (\theta_{\text{sh}}; \theta_1) - \nabla_{\theta_{\text{sh}}} L_2 (\theta_{\text{sh}}; \theta_2) \|_2^2} \right]_+ \\
[a]_+ &= \max(\min(a, 1), 0). \qquad (11)
\end{aligned}$$

According to the chain derivative rule and the multivariate function derivative rule, in Step B, when calculating the optimal $\hat{\beta}$ in (9), we can set θ_{sh}^b to be the weight of last fully connected (FC) layer of the regressor, L_1 to be the L_S, and L_2 to be the L_{hm}; in Step C, when calculating the optimal $\hat{\nu}$ in (9), we can set θ_{sh}^c to be the weight of final FC layer of the feature extractor, L_1 to be the L_{hm}, and L_2 to be the L_{mmd}.

For more information about the derivation process of MGDA, interested readers may refer to [37]. The overall procedure is summarized in Algorithm 1.

Algorithm 1 TRAIL Algorithm

```
Data: Source domain data: $\mathcal{D}_S = \{(\mathbf{x}_S^i, \boldsymbol{y}_S^i)\}_{i=1}^{n_S}$;
      Target domain data: $\mathcal{D}_T = \{(\mathbf{X}_T(j, :, :))\}_{j=1}^{n_T}$;
      fingerprints to be located $\bar{\mathbf{x}}$;
Input: max epoch $n$;
Output: The predicted position of $\bar{\mathbf{x}}$ : $\bar{\boldsymbol{y}}$;
Step A:
Train the model using (4) with $\mathbf{X}_S$
$i = 0$
while $i < n$ do
    Step B:
    Compute $L_S$, $L_{hm}$ in (6)
    Compute $\beta$ in (9) using (11)
    Compute $L_B$ in (9)
    Freeze feature extractor and update regressor using $L_B$
    Step C
    Compute $L_S$, $L_{hm}$, $L_{mmd}$ in (7)
    Compute $\nu$ in (9) using (11)
    Compute $L_C$ in (9)
    Freeze regressor and update feature extractor using $L_C$
    $i = i + 1$;
end while
Return the predicted position $\bar{\boldsymbol{y}}$ of $\bar{\mathbf{x}}$
```

IV. Experiment

In this section, we describe the experimental datasets and settings, show the detailed results, and present our discussions.

A. Datasets

We first introduce two datasets A and B used in our experiment. Table II presents more details about dataset A and dataset B.

TABLE II
DESCRIPTION OF THE TWO DATASETS

Datasets	Feature	Size(m^2)	RPs	BSs	Features	Offline samples	Target samples	Duration
A	RSS	308.4	48	620	200(coded)	11520	3456(each month)	25 months
B	RSS	400	400	16	16	2400	2400	-

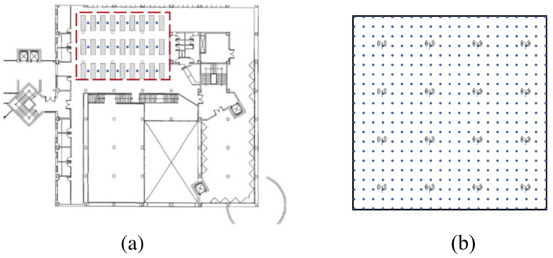


Fig. 4. Experimental environments of the two datasets showing the RP locations (blue circles). (a) Floor plan of dataset A. The region in the dashed red box shows the testbed environment. (b) Floor plan of dataset B.

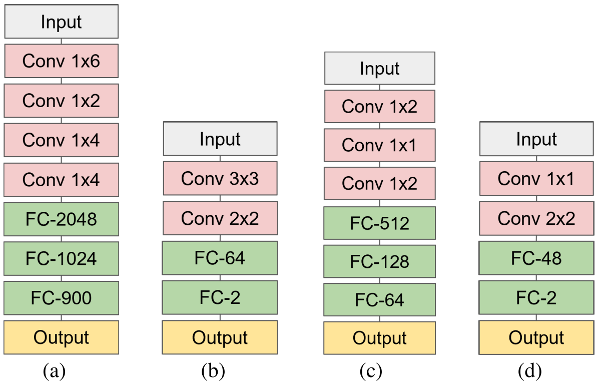


Fig. 5. Network structure for dataset A and dataset B (Conv is the convolution layer followed by the convolution kernel size. FC is the fully connected layer followed by the number of nodes). (a) Feature extractor for dataset A. (b) Regressor for dataset A. (c) Feature extractor for dataset B. (d) Regressor for dataset B.

1) *Dataset A:* UJI dataset, which is a WiFi RSS fingerprint dataset collected on two floors (third and fifth) of the library at University Jaume I in Spain, covered an area of 308.4 m^2 and spanned 25 months, as shown in Fig. 4(a). This dataset suffers from dynamic environmental changes and the combined effect of different interferences. The data for the first month has 11 520 fingerprints, each following month has 3456 fingerprints, and the last month has 6912 fingerprints. At each RP, the measurer performed six measurements of RSS values. The last five measurements are considered valid data because of the residual influence from the prior RP measurement. During the 25 months of the experiment, a total of 620 different BSs appeared in the environment, with significant changes after the 11th month. For detailed information about the dataset, readers may refer to [12].

2) *Dataset B:* To further examine the localization performance of TRAIL under high-noise conditions quantitatively, we conduct simulations of the indoor channel environment using the path loss lognormal shadowing model [38]

$$\mathrm{RSS}^c = \mathrm{RSS}_0^c - 10\gamma \log_{10}\left(\frac{d_c}{d_0}\right) + n_c + n_e \qquad (12)$$

where RSS^c (in dBm) is the RSS measured at the RP from the cth BS, RSS_0^c is the RSS from the cth BS at the reference distance, d_0 and d_c denote the distance between the cth BS and the RP, γ is the path loss exponent (PLE), and n_c indicates the zero-mean normal random variable with variance σ_c^2. To distinguish the source and target domain, we simulate two different hardware by adding additional noise n_e. The mean of squared distance over noise variance is how we calculate the SNR in this context, expressed in decibels (dB), i.e., $\sigma_c^2 = (d_c^2/10^{\mathrm{SNR}/10})$. Additionally, we assume that if the RSS value is equal to or lower than -100 dBm, the BS cannot be detected. Specifically, we simulate a 20 × 20 m indoor environment with 16 BSs evenly distributed, as shown in Fig. 4(b). Then, we divide it into 400 RPs spaced 1 m apart and collect six measurements at each RP, resulting in 2400 fingerprints. We set $\gamma = 2$, sample n_e from a zero-mean Gaussian distribution with a variance of 25, and vary the SNR from 7 to 20 dB to create 14 pairs of source and target domains.

B. Experimental Settings

We use the average location error (ALE) distance as the evaluation metric, which is defined as follows:

$$\mathrm{ALE} = \frac{1}{n}\sum_{i=1}^{n} \|\hat{\boldsymbol{y}}^i - \boldsymbol{y}^i\|_2 \qquad (13)$$

where n is the number of fingerprints, $\hat{\boldsymbol{y}}^i$ is the predicted position, and $\boldsymbol{y}^i$ is the true position.

We compare TRAIL with the following transfer learning methods: TCA [17], DAN [19], DeepCoral [15], and TransLoc [20]. To compare TRAIL with adversarial-based methods, we consider DANN [25], MCD [28], and iToLoc [30]. We used the same experimental parameters in all our methods, i.e., 0.001 for learning rate, 128 for batch size, 300 for maximum epoch, and 20 for early stopping. Besides, because the two datasets have disparate data dimensions, we use different deep neural network structures for TRAIL, as shown in Fig. 5. We do not employ TransLoc [20] on dataset B because the fingerprint of dataset B has no dimensional heterogeneity problem.

TABLE III
ALEs for Different Methods

Method	ALE(m)	
	Dataset A	Dataset B
DAN	2.69	0.71
DeepCoral	2.71	0.72
DANN	2.38	0.73
MCD	2.62	0.79
TCA	3.67	1.01
TransLoc	2.25	None
iToLoc	2.84	0.97
TRAIL-onestep	2.51	None
TRAIL	**1.77**	**0.57**

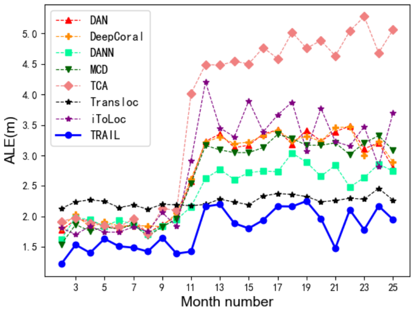


Fig. 6. ALEs of all methods on dataset A.

In the experiments on dataset A, we utilize all the data from the first month as the source domain, and the remaining months serve as the target domain. For each domain adaptation task, we consider out-of-sample setting, i.e., the model is trained using the first and 80% of the lth month data (2764 samples) and then used for prediction on the same lth month unseen test data (692 samples). Because the fingerprint matrix of dataset A is sparse, we train an autoencoder to convert the original 620-D fingerprint to a 200-D code and apply it to all methods, except TransLoc [20] and iToLoc [30].

In the experiments on dataset B, since the simulated environment only has 16 BSs, we do not use the autoencoder to reduce the data dimension. Instead, we use the source domain and 80% of the target domain data (1120 samples) for each SNR as the training set and the remaining 20% of the target domain data as the test set (280 samples).

C. Localization Performance

Table III illustrates the ALEs for all months using dataset A and for all SNRs using dataset B. Fig. 6 shows the ALEs of different methods for each month using dataset A. We observe average ALEs of 3.67 m for TCA [17], 2.69 m for DAN [19], 2.38 m for DANN [25], 2.71 m for DeepCroal [15], 2.25 m for TransLoc [20], 2.62 m for MCD [28], and 2.54 m for iToLoc [30]. In contrast, TRAIL achieves an ALE of 1.77 m, which is at least 21% better than the others. The TCA method shows the worst performance, probably due to its limited fitting ability, as it is not a deep learning method. The DANN method performs unsatisfactorily in this experiment as the training of the domain discriminator requires a large amount of data. The DAN method only optimizes the MMD distance, leading to an inferior performance compared to TRAIL.

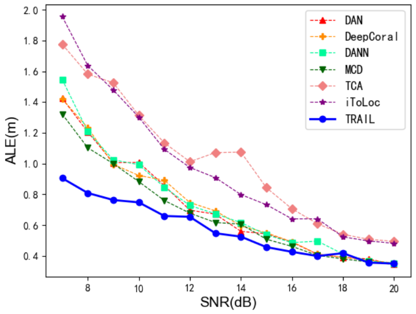


Fig. 7. ALEs of all methods on dataset B.

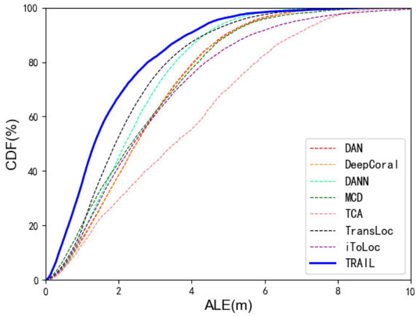


Fig. 8. CDF of ALEs on dataset A.

With the change of BSs after the 11th month in dataset A, there is a considerable decrease in the similarity between the source and target domain data, resulting in a significant increase in ALE for most of the methods. Because TransLoc is the only method that considers the change of BSs, it has relatively stable ALEs for all months. Although TRAIL has an uptick in ALE after the 11th month, it noticeably outperforms all other methods, demonstrating its superiority for domain adaptation with large discrepancy and robustness for long-term indoor localization. Furthermore, in the first ten months, our strategy also surpasses all others with compelling results, under the condition that more considerable cross-domain similarity leads to close ALEs for all other approaches.

Fig. 7 shows the localization performance on the simulated dataset B. Except for TCA and iTocLoc, all other methods produce comparable results in the case of a high SNR. Besides, the superiority of TRAIL becomes more significant under low SNR conditions. When SNR = 7 dB, TRAIL achieves an ALE of 0.9 m, significantly outperforming the other methods.

As shown in Fig. 8, the cumulative distribution function (cdf) curve of TRAIL outperforms the cdf curves of other methods on dataset A. Table IV illustrates the percentage of samples with ALE less than 2 m. We can observe that the percentage of samples with ALE less than 2 m of TRAIL is 67.31%, outperforming all the comparable methods. This is 14.76% higher than TransLoc and 37.74% higher than TCA. The cdf of all methods on dataset B when SNR = 7 dB is depicted in Fig. 9. TRAIL reduces the 90th percentile ALE by 53%, 55%, 48%, 47%, 40%, and 62% as compared with

TABLE IV
PERCENTAGE OF SAMPLES WITH ALE LESS THAN 2 m

Method	Percentage(%)	
	Dataset A	Dataset B(SNR=7dB)
DAN	38.50	83.33
DeepCoral	38.60	82.70
DANN	44.98	77.91
MCD	43.17	77.29
TCA	29.57	70.41
TransLoc	52.55	None
iToLoc	41.61	70.02
TRAIL	**67.31**	**96.25**

TABLE V
ALES OF TRAIL WITH DIFFERENT LOSS WEIGHTS

	β, ν	ALE(m)
Fixed weight	0.1	2.87
	0.01	1.89
	0.001	2.03
TRAIL	Self-learning	**1.77**

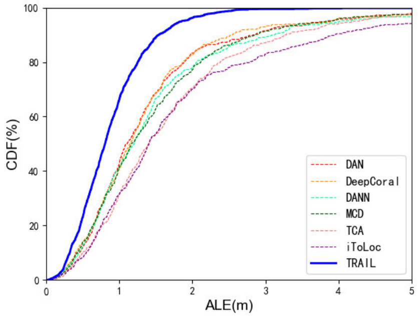


Fig. 9. CDF of ALEs on dataset B (SNR = 7 dB).

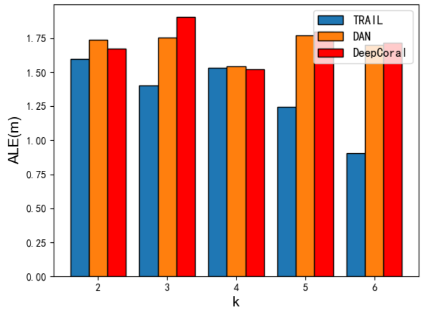


Fig. 10. Effect of different k (dataset B, SNR = 7 dB).

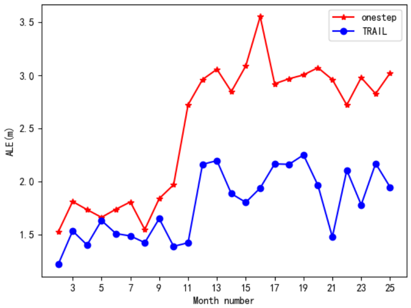


Fig. 11. Comparison between TRAIL and one step training (dataset A).

DANN, TCA, MCD, DAN, DeepCroal, and iToLoc, respectively. Table IV also illustrates the percentage of samples with ALE less than 2 m when SNR = 7 dB. We can observe that the percentage of samples with ALE less than 2 m of TRAIL is 96.25%, while that of DANN, TCA, MCD, DAN, DeepCroal, and iToLoc are 77.91%, 70.41%, 77.29%, 83.33%, 82.70%, and 70.02%, respectively.

Comparing the results in Figs. 6 and 7, when SNR is large without other unknown interference (SNR = 20 dB in Fig. 7), the ALEs of TRAIL and other methods are very close, which indicates that TRAIL has similar effects as other methods in solving the differences in domain distribution caused by dynamic changes in the indoor environment and equipment heterogeneity. Increasing the model's tolerance to unknown interference will not destroy the original data's separability. In addition, TRAIL is much more effective than other methods in large SNR and real-world environments, thanks to using the adversarial method to eliminate the combined effects of different noises during the online phase. The experimental results on dataset A and dataset B show that TRAIL simultaneously deals with the two error factors in indoor fingerprint positioning tasks in an effective way. TRAIL consistently achieved the best results on both dataset A and dataset B, which certainly demonstrates its universality in a variety of indoor environments.

To further explore the influence of the number of hetero-measure samples on positioning results, we conducted experiments on different k in (2) on dataset B with SNR = 7 dB, as shown in Fig. 10. It can be found that with the increase of k, the ALE of TRAIL presents a downward trend while the ALEs of other methods almost hold steady, which shows that TRAIL can utilize the differences and connections between hetero-measure samples to improve positioning accuracy.

D. Effectiveness of Three-Step Adversarial Training

To validate the effectiveness of the three-step adversarial training, we introduced a gradient inversion layer between the feature extractor and the regressor, allowing L_{hm} to take its opposite during backpropagation. Subsequently, we trained two models using the three-step adversarial and one-step training methods. The localization outputs are presented in Fig. 11. The three-step adversarial and one-step training strategies achieve ALEs of 1.77 and 2.51 m on dataset A, respectively. These results indicate that the three-step adversarial training method outperforms the one-step training method by a significant margin of 40%.

E. Effectiveness of MDGA

To analyze the effect of different loss weights on localization results, we repeat the experiments on dataset A using the parameters $\beta = \nu = 0.1$, 0.01, and 0.001 in (9). As shown

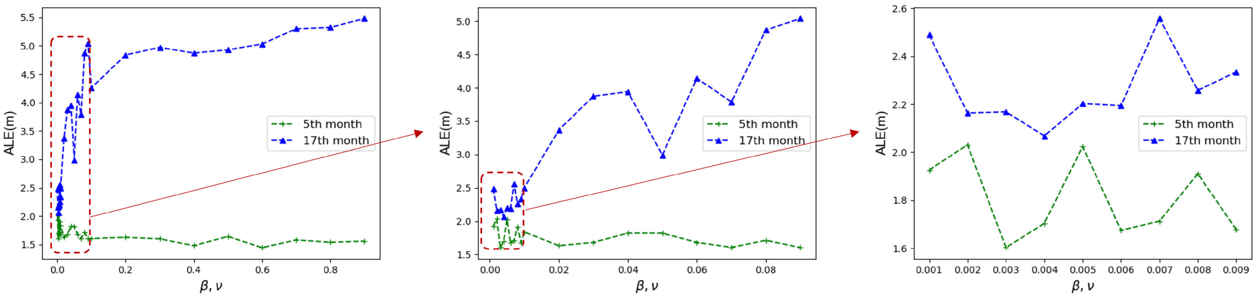


Fig. 12. Effect of different fixed weights (dataset A).

in Table V, TRAIL with MDGA achieves an ALE of 1.77 m, while that of fixed weight 0.1, 0.01, 0.001 are 2.87, 1.89, and 2.03 m, respectively.

To further demonstrate the limitation of fixed weight, we conduct experiments using the fifth and 17th months' data in dataset A. In detail, we systematically vary the parameters β and ν in (9) over the ranges [0.001, 0.009] in increments of 0.001, [0.01, 0.09] in increments of 0.01, and [0.1, 0.9] in increments of 0.1. The experiment results indicate that the change in loss weights significantly impacts TRAIL's positioning accuracy. As demonstrated in the third plane of Fig. 12, a 0.001 variation of β and ν can result in an ALE difference of up to 0.4 m. This effect is even more pronounced in the 17th month, where changing the β and ν from 0.04 to 0.05 leads to a decrease of 1 m in ALE. Furthermore, the effect of the same β and ν on different months is inconsistent. For example, with $\beta = \nu = 0.007$, a satisfactory ALE is achieved in the fifth month, whereas a terrible ALE is performed in the 17th month. With the increase of β and ν, the ALEs in the 17th month show an overall upward trend, while the ALEs in the fifth month are relatively stable. This is because the positioning error is mainly caused by the dynamic change of the indoor environment and the comprehensive interference in the online stage, which is tackled by L_{mmd} and L_{hm} respectively in TRAIL. However, increasing the weight of L_{hm} will affect the transfer effect while decreasing the weight of it will indulge the online synthesis interference. For the fifth and 17th months' data in dataset A, the fifth month's data are more similar to the source domain (the first month) data, so transferring source domain knowledge will result in more accurate positioning. Therefore, its positioning error increases as the β and ν increase. For the data of the 17th month, its similarity to the source domain data is slight, so the weights less influence its positioning error.

These findings indicate that the weights need to be carefully selected according to the similarity between source and target domain data, which means it is impossible to choose a fixed weight suitable for all months based on empirical observations. Fortunately, the MGDA can effectively adapt to the changes in different domain data.

V. Conclusion

In this article, we proposed TRAIL, which can alleviate the problem of distribution discrepancy between online and offline data and differences among hetero-measure samples for fingerprint-based indoor localization. Specifically, in Step A, we pre-trained the model with offline data to learn the underlying association between offline RSS data and locations. In Step B, we refined the regressor's capacity to identify more intricate "*hard*" samples. Finally, we trained the feature extractor to learn a suitable transformation to map the data into a feature space with slight domain discrepancy and to the "*soft*" points of the regression hyperplane. Experiments on both actual and simulated datasets have validated the effectiveness of TRAIL. Additionally, TRAIL can be applied not only with MKMMD but also with other transfer learning methods, which deserves to be explored in future research.

References

[1] 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., 2019.

[2] B. Huang, L. Xie, and Z. Yang, "TDOA-based source localization with distance-dependent noises," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 468–480, Jan. 2015.

[3] Y. Li, Z. He, Z. Gao, Y. Zhuang, C. Shi, and N. El-Sheimy, "Toward robust crowdsourcing-based localization: A fingerprinting accuracy indicator enhanced wireless/magnetic/inertial integration approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3585–3600, Apr. 2019.

[4] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM Conf. Comput. Commun. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, vol. 2, Mar. 2000, pp. 775–784.

[5] X. Guo, N. Ansari, L. Li, and L. Duan, "A hybrid positioning system for location-based services: Design and implementation," *IEEE Commun. Mag.*, vol. 58, no. 5, pp. 90–96, May 2020.

[6] A. Shahmansoori, G. E. Garcia, G. Destino, G. Seco-Granados, and H. Wymeersch, "Position and orientation estimation through millimeter-wave MIMO in 5G systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1822–1835, Mar. 2018.

[7] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst. Appl. Services*, 2005, pp. 205–218.

[8] R. W. Ouyang, A. K. Wong, C.-T. Lea, and M. Chiang, "Indoor location estimation with reduced calibration exploiting unlabeled data via hybrid generative/discriminative learning," *IEEE Trans. Mobile Comput.*, vol. 11, no. 11, pp. 1613–1626, Nov. 2012.

[9] X. Guo, N. R. Elikplim, N. Ansari, L. Li, and L. Wang, "Robust WiFi localization by fusing derivative fingerprints of RSS and multiple classifiers," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3177–3186, May 2020.

[10] J. Niu, B. Wang, L. Cheng, and J. J. P. C. Rodrigues, "WicLoc: An indoor localization system based on WiFi fingerprints and crowdsourcing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 3008–3013.

[11] H. Si, X. Guo, and N. Ansari, "Multi-agent interactive localization: A positive transfer learning perspective," *IEEE Trans. Cognit. Commun. Netw.*, early access, Nov. 6, 2023, doi: 10.1109/TCCN.2023.3330062.

[12] G. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E. Lohan, and J. Huerta, "Long-term WiFi fingerprinting dataset for research on robust indoor positioning," *Data*, vol. 3, no. 1, p. 3, Jan. 2018.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.

[14] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, Jul. 2006.

[15] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 443–450.

[16] C. Shui, Z. Li, J. Li, C. Gagné, C. X. Ling, and B. Wang, "Aggregating from multiple target-shifted sources," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9638–9648.

[17] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2010.

[18] Z. Sun, Y. Chen, J. Qi, and J. Liu, "Adaptive localization through transfer learning in indoor Wi-Fi environment," in *Proc. 7th Int. Conf. Mach. Learn. Appl.*, 2008, pp. 331–336.

[19] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 97–105.

[20] L.Lin, X. Guo, M. Zhao, H. Li, and N. Ansari, "TransLoc: A heterogeneous knowledge transfer framework for fingerprint-based indoor localization," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3628–3642, Jun. 2021.

[21] L. Li, X. Guo, Y. Zhang, N. Ansari, and H. Li, "Long short-term indoor positioning system via evolving knowledge transfer," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5556–5572, Jul. 2022.

[22] X. Guo, L. Li, F. Xu, and N. Ansari, "Expectation maximization indoor localization utilizing supporting set for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2573–2582, Apr. 2019.

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *Stat*, vol. 1050, pp. 9–36, Jun. 2017.

[24] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680.

[25] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1–35, 2016.

[26] Z. Du, J. Li, H. Su, L. Zhu, and K. Lu, "Cross-domain gradient discrepancy minimization for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3937–3946.

[27] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7472–7481.

[28] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732.

[29] X.-Y. Liu and X. Wang, "Real-time indoor localization for smartphones using tensor-generative adversarial nets," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3433–3443, Aug. 2021.

[30] D. Li, J. Xu, Z. Yang, Y. Lu, and X. Zhang, "Train once, locate anytime for anyone: Adversarial learning based wireless localization," in *Proc. IEEE INFOCOM*, May 2021, pp. 1–10.

[31] M. Zhang, Z. Fan, R. Shibasaki, and X. Song, "Domain adversarial graph convolutional network based on RSSI and crowdsensing for indoor localization," *IEEE Internet Things J.*, vol. 10, no. 15, pp. 13662–13672, Mar. 2023.

[32] Y. Zhao, Y. Liu, T. He, A. V. Vasilakos, and C. Hu, "FREDI: Robust RSS-based ranging with multipath effect and radio interference," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 505–509.

[33] S.-H. Fang, T.-N. Lin, and K.-C. Lee, "A novel algorithm for multipath fingerprinting in indoor WLAN environments," *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3579–3588, Sep. 2008.

[34] X. Fang and L. Chen, "An optimal multi-channel trilateration localization algorithm by radio-multipath multi-objective evolution in RSS-ranging-based wireless sensor networks," *Sensors*, vol. 20, no. 6, p. 1798, Mar. 2020.

[35] M. N. Husen and S. Lee, "Indoor human localization with orientation using WiFi fingerprinting," in *Proc. 8th Int. Conf. Ubiquitous Inf. Manage. Commun.*, Jan. 2014, pp. 1–6.

[36] J. Bi, Y. Wang, X. Li, H. Cao, H. Qi, and Y. Wang, "A novel method of adaptive weighted K-nearest neighbor fingerprint indoor positioning considering user's orientation," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 6, Jun. 2018, Art. no. 155014771878588.

[37] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 527–538.

[38] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo, "Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure," *Wireless Netw.*, vol. 16, no. 2, pp. 405–420, Feb. 2010.

Yin Yang was born in Sichuan, China, in 1999. He received the B.Eng. degree from the School of Electronic Information, Sichuan University, Chengdu, China, in 2021. He is currently pursuing the M.Eng. degree in communication with the University of Electronic Science and Technology of China, Chengdu.

His current research interests include machine learning, indoor localization, adversarial learning, and transfer learning.

Xiansheng Guo (Senior Member, IEEE) received the B.Eng. degree from Anhui Normal University, Wuhu, China, in 2002, the M.Eng. degree from the Southwest University of Science and Technology, Mianyang, China, in 2005, and the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2008.

From 2008 to 2009, he was a Research Associate at the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong. From 2012 to 2014, he was a Research Fellow at the Department of Electronic Engineering, Tsinghua University, Beijing, China. From 2016 to 2017, he was a Research Scholar at the Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Professor at the Department of Electronic Engineering, UESTC. His research interests include array signal processing, wireless localization, machine learning, information fusion, and software radio design.

Cheng Chen received the B.Eng. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2006, and the master's degree in computer science and technology from Tsinghua University, Beijing, China, in 2011, where he is currently pursuing the Ph.D. degree.

His research interests include intelligent transportation systems (ITS), automated valet parking (AVP), and machine learning.

Gordon Owusu Boateng (Member, IEEE) received the bachelor's degree in telecommunications engineering from the Kwame Nkrumah University of Science and Technology (KNUST), Kumasi, Ghana, in 2014, and the M.Eng. and Ph.D. degrees in computer science and technology from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2019 and 2023, respectively.

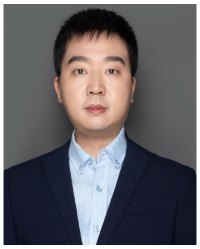
From 2014 to 2016, he worked under sub-contracts at Ericsson, Accra, Ghana; and TIGO, Accra. He is currently a Postdoctoral Researcher at the Hybrid Positioning Research Group (HPRG), UESTC. Till now, he has published over 25 scientific journals and conference papers. His research interests include 5G/6G wireless networks, blockchain, reinforcement learning, vehicular networks, target positioning, and automated valet parking.

Haonan Si (Graduate Student Member, IEEE) was born in Jiangsu, China, in 1999. He is currently pursuing the Ph.D. degree in communication with the University of Electronic Science and Technology of China, Chengdu, China.

His current research interests include machine learning, fuzzy logic systems, indoor localization, and neural learning control.

Bocheng Qian was born in Jiangxi, China, in 1999. He is currently pursuing the Ph.D. degree in communication with the University of Electronic Science and Technology of China, Chengdu, China.

His current research interests include machine learning, multi-modal data fusion, indoor localization, and magnetic field positioning.



Linfu Duan received the B.Eng., M.Eng., and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009, 2012, and 2018, respectively.

He is currently a Research Fellow at UESTC. His research interests include wireless positioning and ultra-wideband positioning.