Towards Robust WiFi Fingerprint-based Vehicle Tracking in Dynamic Indoor Parking Environments: An Online Learning Framework

Kai Liu, *Senior Member, IEEE*, Feiyu Jin, Junbo Hu, Ruitao Xie, Fuqiang Gu, *Member, IEEE*, Songtao Guo, *Senior Member, IEEE*, and Jiangtao Luo, *Senior Member, IEEE*

Abstract—The variation of wireless signal in dynamic indoor parking environments may seriously compromise the performance of fingerprint-based localization methods. In this regard, this paper investigates the problem of robust WiFi fingerprint-based vehicle tracking in dynamic indoor parking environments, aiming at designing an online learning framework to continuously train the localization model and counteract the effect of signal variation. Specifically, a Hidden Markov Model (HMM) based Online Evaluation (HOE) method is firstly proposed to assess the accuracy of localization results by measuring the inconsistency of locations inferred by WiFi fingerprinting and Dead Reckoning (DR). Further, an Online Transfer Learning (OTL) algorithm is designed to improve the robustness of the fingerprinting localization, which consists of a weight allocation scheme to combine two classification models (i.e., the batch model and the online model) and an instance-based transferring scheme to resample the offline fingerprints and retrain the batch model. Finally, we implement the system prototype and give comprehensive performance evaluation, which demonstrates that the proposed solutions can outperform the state-of-the-art localization algorithms around 28% $\sim$ 58% on vehicle tracking accuracy in dynamic indoor parking environments.

Index Terms—Vehicle Tracking, Indoor Localization, WiFi Fingerprint, Hidden Markov Model, Online Learning

1 Introduction

Efficient indoor vehicle tracking is a key enabler of emerging applications such as indoor parking navigation and real-time parking coordination [1]. In general, current indoor localization technologies can be classified into three categorizes, i.e., Dead Reckoning (DR)-based, ranging-based and fingerprinting-based schemes. Nevertheless, existing solutions cannot well support vehicle tracking in dynamic indoor parking environments due to either high hardware/computation cost or low system scalability/robustness [2] [3]. In view of this, this work aims at designing an online learning framework to enhance the robustness of WiFi fingerprinting localization and enable efficient vehicle tracking in dynamic indoor parking environments.

WiFi fingerprinting is one of the promising solutions to indoor localization due to the prevalence and low cost of WiFi device [4] [5]. Specifically, during the offline training phase, a site-survey is conducted to build the fingerprint database by collecting the Received Strength Signal Indicator (RSSI) features (i.e., WiFi Fingerprints) from ambient Access Points (APs) at Reference Points (RPs) with known coordinates. During the online localization phase, real-time measured RSSIs are categorized into different RPs based on the fingerprint database and certain classification algorithm, such as K-Nearest-Neighbor (KNN). Apparently, such a technique has the advantage of wide availability and easy implementation [6].

However, WiFi RSSI features may change over time due to the multipath effect and dynamic environments, which undermines the robustness of the localization system [7]. Some studies designed sophisticated representation of RSSI features based on machine learning technologies such as feature scaling [8] and stacked denoising autoencoder [9]. However, the RSSI features may still deviate from the training set due to the change of physical layout or the long-term environmental factors such as temperature and humidity [10]. Some other solutions utilized WiFi sniffers [11] or crowdsourcing [12], aiming at compensating for the impact of dynamic environments by periodically updating the fingerprint database. Nevertheless, it is still challenging to enable robust vehicle tracking in dynamic indoor parking environments. First, the different layout of parked vehicles as well as the mobility of tracking vehicles lead to highly dynamic changing of RSSI features. Therefore, it would cause exhaustive updating overheads for conventional fingerprint-updating based methods, resulting in low system scalability. Moreover, current crowdsourcing-based methods are not practical in indoor parking environments, because different participants may sample quite inconsistent RSSI features across different timescales, which may cause

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 62172064, 62171072, 61872049, and 61802263, and in part by the Chongqing Young-Talent Program (Project No. cstc2022ycjh-bgzxm0039). (Corresponding author: Ruitao Xie)

K. Liu is with the College of Computer Science, Chongqing University, Chongqing 400040, China, and also with School of Electrical and Computer Engineering, Nanfang College, Guangzhou, China. (e-mail: liukai0807@cqu.edu.cn).

Feiyu Jin, Junbo Hu, Fuqiang Gu, Songtao Guo are with the college of Computer Science, Chongqing University, Chongqing, 400040, China. (email: fyjin, junbohu, gufq, guosongtao@cqu.edu.cn)

Ruitao Xie is with the College of Computer Science and Software Engineering, Shenzhen University, Guangdong, 518060, China (email: drtxie@gmail.com)

Jiangtao Luo is with the Electronic Information and Networking Research Institute, Chongqing University of Posts and Telecommunications, Chongqing, 400065 (email: Luojt@cqupt.edu.cn)

serious deterioration of the localization performance.

With above motivations, this work makes the first effort on proposing an online learning framework as well as tailored solutions to enable robust WiFi fingerprint-based vehicle tracking in dynamic indoor parking environments. The main contributions of this work are summarized as follows:

- We design an online learning framework, which contains a Hidden Markov Model (HMM) based Online Evaluation (HOE) method and an Online Transfer Learning (OTL) algorithm. First, the WiFi fingerprinting localization scheme in OTL predicts the localization result based on online RSSI samples. Then, the HOE assesses the accuracy of WiFi fingerprinting localization results and calibrates the vehicle's location. In turn, the OTL will construct candidate label sets for online RSSI samples and update the corresponding parameters of the localization model based on the evaluation results, which can further improve the robustness for vehicle tracking in dynamic indoor parking environments.
- For HOE, first, an HMM is derived to measure the consistency of locations estimated by WiFi fingerprinting and DR as the evaluation results. In HMM, the emission probability is modelled based on WiFi fingerprinting localization results with the Gaussian distribution and the transition probability is modelled based on the displacement estimated by DR via Monte Carlo Simulation (MCS). Then, a medium probability is derived to represent vehicles' position distribution after transition, and a forward probability is derived to estimate the final distribution of vehicle's locations. Finally, the Wasserstein distance is adopted to measure the similarity between the medium probability and the emission probability, which indicates the accuracy of the WiFi fingerprinting localization results.
- For OTL, first, a weight allocation scheme is designed to combine two classification models, including the batch model and the online model, which are trained by the offline fingerprints and the online signal features, respectively. It is proved that with N rounds of weight allocation, the loss of the WiFi fingerprint localization after the combination is bounded by $\sqrt{2N\ln 2}+\ln 2$ of the best base model. Then, an instance-based transferring scheme is proposed to continuously update the fingerprint database by resampling offline fingerprints based on their similarities to the current RSSI features. It is proved that that through M rounds of instance-based transferring the loss of offline fingerprint database with total $\mathcal{N}_f$ fingerprints is bounded by $\sqrt{2M\ln\mathcal{N}_f}+\ln\mathcal{N}_f$ of the minimum loss of an individual fingerprint.
- We build the system prototype and give an extensive performance evaluation in real-world indoor parking environments. Specifically, an android-based APP is implemented and placed in vehicles to collect the WiFi RSSI samples along with inertial sensors' readings. We have driven over 20 kilometers in an indoor parking area to collect the data, which are used to train the algorithm parameters and testing the algorithm performance. Existing competitive algorithms have been implemented for performance comparison and the results conclusively demonstrate the superiority of the proposed solution on enhancing vehicle tracking performance in dynamic indoor parking environments.

The rest of this paper is organized as follows. Section II designs the online learning framework. Section III proposes the HOE, and Section IV proposes the OTL. In Section V, we build the system prototype and carry out experiments. Section VI reviews the related work. Finally, we conclude this paper and discuss future directions in Section VII.

2 The Online Learning Framework

This section presents an online learning framework to address the following two challenges. First, it is nontrivial to label the online RSSI samples with their corresponding truth labels, which is critical to make the framework be adapted to the dynamic environment. Second, it is important yet challenging to incorporate the offline fingerprint database in an efficient way since a) it may contain outdated fingerprints due to the changing of RSSI patterns; b) parts of the offline fingerprints might still help if they could be properly combined with the online model, especially when the online training has not yet converged.

The primary components as well as the workflow of the online learning framework are shown in Fig.1. There are two major components, namely, the OTL algorithm and the HOE method. First, the mobile device on the driving vehicle collects the online RSSI samples and the inertial sensors' data. Then, the online RSSI samples are utilized to predict vehicle's locations based on the batch and online classification models. On the other hand, the inertial sensors' data are utilized to estimate the displacement of vehicle based on DR. On this basis, the fingerprinting localization results and the displacement are input into HOE, where the fingerprinting localization results are further calibrated and evaluated. In turn, the evaluation outcomes are sent to OTL for updating the online model and the corresponding parameters. Primary functions of HOE and OTL are outlined as follows.

(1) The HOE method: It is designed to evaluate the WiFi fingerprinting localization results. In particular, if two consecutive WiFi fingerprinting localization results are aligned with the displacement estimated by DR, then they are considered as accurate. Then, the labels predicted by fingerprinting localization model are considered as valid labels for online training. To realize such evaluation, an HMM is established based on the WiFi fingerprinting localization results and the displacement estimated by DR, which measures the alignment of these two methods and further enhances the estimation of vehicle's locations.

(2) The OTL algorithm: It is designed to make the vehicle tracking be adapted to the dynamic environment by exploring the synergistic effect of offline fingerprints and online RSSI samples. Specifically, if the fingerprinting localization results were accurate, the label conversion

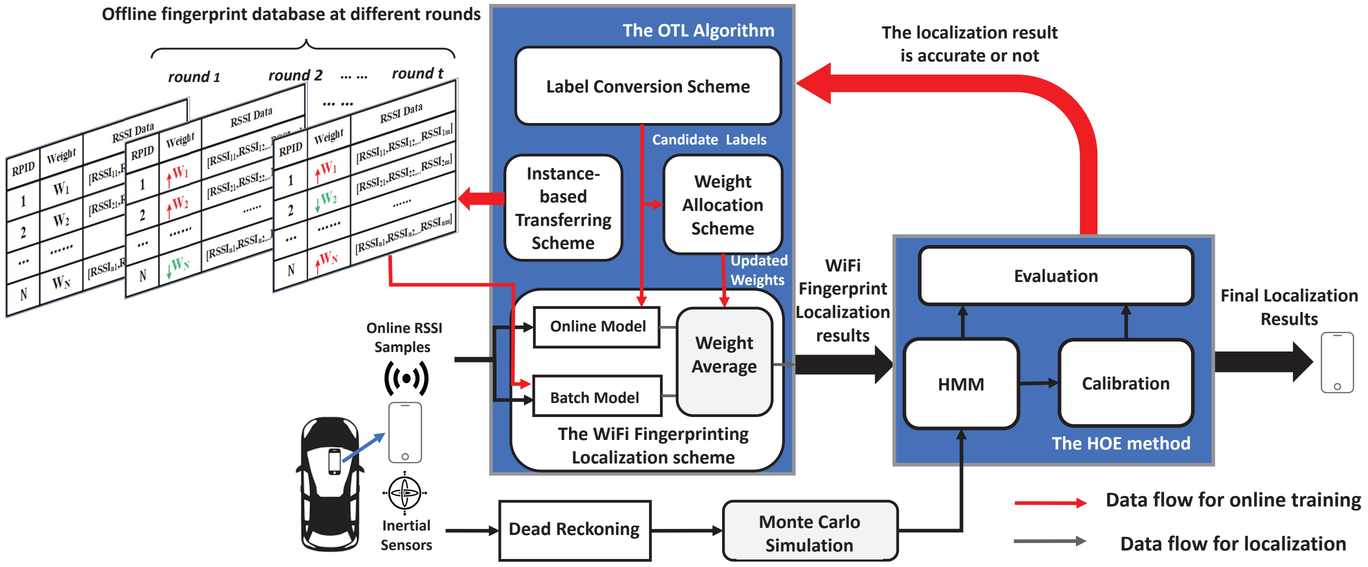


Fig. 1. The proposed online learning framework

scheme would construct a candidate label set for online RSSI samples to train the online model, making it more suitable for the current wireless environment. In addition, although parts of the offline data may have outdated due to the changing of RSSI features, some of the fingerprints may be affected less than others, which can be transferred to train the batch model. Therefore, for those offline fingerprints which can still reflect current signal features, an instance-based transferring is conducted to form a new offline fingerprint database to further retrain the batch classification model. On this basis, the two classification models are collaborated to estimate the vehicle's locations by assigning proper weights during online trials.

In the following, we design the HOE and the OTL in details. The primary notations are summarized in Table 1.

3 HMM BASED ONLINE EVALUATION METHOD

In this section, we first derive an HMM to model the WiFi fingerprinting localization results and the displacement estimated by DR. Then, an online calibration and evaluation method is proposed to calibrate the fingerprinting localization results and examine the inconsistency of the locations inferred by WiFi fingerprinting and DR.

3.1 A Hidden Markov Model

The localization area is divided into small square grids and each grid is considered as a hidden state in HMM. The set of hidden states is represented by $(g_1, g_2 \dots, g_{\mathcal{N}_g})$, where $\mathcal{N}_g$ is the number of grids. The coordinate of the centroid of g_i is denoted by $\mathbf{p}_i^{(g)} = \left(x_i^{(g)}, y_i^{(g)}\right)$. The WiFi fingerprinting localization result $\mathbf{z}_t$ at round t is considered as the state measurement and it is used to model the emission probability, which is represented by $\mathbf{B}(\mathbf{z}_t) = \{b_i(\mathbf{z}_t) \mid 1 < i < \mathcal{N}_g\}$, where $b_i(\mathbf{z}_t) = P(\mathbf{z}_t \mid g_i)$. The emission probability gives the likelihood that the localization result $\mathbf{z}_t$ would be observed if the vehicle was actually on grid g_i. The displacement of the vehicle inferred by DR between two consecutive fingerprinting localization rounds is used to model the transition probability, which is denoted by $\mathbf{A}_t = \left\{A_t^{ij} \mid 1 < i < \mathcal{N}_g, 1 < j < \mathcal{N}_g\right\}$, where $A_t^{ij} = P(g_i \mid g_j)$. The transition probability gives the likelihood of a vehicle moving from grid g_j to g_i during two consecutive fingerprinting localization rounds. The initial state probability is denoted by $\boldsymbol{\pi} = \{\pi_1, \pi_2, \dots, \pi_{\mathcal{N}_g}\}$, where $\pi_i = P(g_i)$, which indicates the probability of vehicle's initial grid. The parameters of HMM are derived as follows.

For initial state probability, it is initialized by the Uniform distribution, since we assume that is no prior knowledge on vehicles' initial positions. Therefore, the π_i of each grid is $1/\mathcal{N}_g$.

The emission probability of each grid $b_i(\mathbf{z}_t)$ is determined by its distance to the WiFi fingerprinting localization results $\mathbf{z}_t$. Given a grid g_i and a localization result $\mathbf{z}_t$, the distance between them is $\left\|\mathbf{z}_t - \mathbf{p}_i^{(g)}\right\|_2$, where $\|\cdot\|_2$ represents the l_2 norm. The error of fingerprinting localization is modelled as a zero-mean Gaussian distribution with the standard deviation δ [13]. Then, the emission probability for a grid g_i is computed by:

$$b_i(\mathbf{z}_t) = P(\mathbf{z}_t \mid g_i) = \frac{1}{\sqrt{2\pi}\delta} e^{-0.5\left(\frac{\left\|\mathbf{z}_t - \mathbf{p}_i^{(g)}\right\|_2}{\delta}\right)^2} \tag{1}$$

The transition probability $\mathbf{A_t}$ is determined by the DR estimated displacement and map constraints. Specifically, the velocity and heading direction of vehicles are extracted from the inertial sensors, which can be used to estimate the

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3200411

TABLE 1
Summary of Primary Notations

Notations	Description	Notes
$\mathbf{p}_t^{(v)}$	calibrated localization result of the vehicle at round t	$\mathbf{p}_t^{(v)}=\left(x_t^{(v)},y_t^{(v)}\right)$
$\mathbf{p}_i^{(g)}$	centroid of grid i	$\mathbf{p}_i^{(g)}=\left(x_i^{(g)},y_i^{(g)}\right)$
$\mathbf{p}_{l_i}^{(rp)}$	location of RP l_i	$\mathbf{p}_{l_i}^{(rp)}=\left(x_{l_i}^{(rp)},y_{l_i}^{(rp)}\right)$
$\mathbf{B}(\mathbf{z}_t)$	emission probability of HMM at round t	$\mathbf{B}\left(\mathbf{z}_t\right)=\left\{b_i\left(\mathbf{z}_t\right)\mid 1<i<\mathcal{N}_g\right\}$
χ_i	sample set of grid i in Monte Carlo Simulation	$\chi_i=\left\{\mathbf{p}_{i1},\mathbf{p}_{i2},\ldots,\mathbf{p}_{i\mathcal{N}_s}\right\}$
$\mathbf{A}_t$	transition probability of HMM at round t	$\mathbf{A}_t=\left\{A_t^{ij}\mid 1<i<\mathcal{N}_g,1<j<\mathcal{N}_g\right\}$
$\boldsymbol{\pi}$	initial state probability of HMM	$\boldsymbol{\pi}=\left\{\pi_1,\pi_2,\ldots,\pi_{\mathcal{N}_g}\right\}$
$\boldsymbol{\alpha}_t$	forward probability at round t	$\boldsymbol{\alpha}_t=\left\{\alpha_t^i\mid 0<i<\mathcal{N}_g\right\}$
$\mathbf{O}_t$	medium probability at round t	$\mathbf{O}_t=\left\{O_t^i\mid 0<i<\mathcal{N}_g\right\}$
R_s	offline fingerprint database	$R_s=\left\{\left(\mathbf{r}_i,l_i\right)\mid i\in\left[1,\mathcal{N}_f\right]\right\}$
$h(\bar{\mathbf{r}}_t)$	classification model	$h\left(\bar{\mathbf{r}}_t\right)=\left\{\mathrm{P}\left(l_j\mid\bar{\mathbf{r}}_t\right)\mid 1<j<\mathcal{N}_{\mathrm{r}}\right\}$
θ_n^o	weight of classification model o for n-th weight allocation	$\theta_n^o\in[0,1]$ where $o=1,2$ and $1\leq n\leq N$
$\mathcal{L}_n^o$	loss of classification model o for n-th weight allocation	$\mathcal{L}_n^o=\llbracket\arg\max_{l_j}h_o\left(\bar{\mathbf{r}}_t\right)\neq l_i^c\rrbracket$
$\mathbf{w}_m$	weights of offline fingerprints for m-th instance-based transferring	$\mathbf{w}_m=\left\{w_m^i\mid 1<i<\mathcal{N}_f\right\}$, where $1\leq m\leq M$
ℓ_m^i	loss of i-th offline fingerprint in m-th instance-based transferring	$\ell_m^i=\left[\left\Vert\mathbf{p}_{l_a}^{(rp)}-\mathbf{p}_{l_i}^{(rp)}\right\Vert_2>\mathcal{T}_S\right]$
$\mathcal{N}_g$	number of grids	
$\mathcal{N}_r$	number of RPs	
$\mathcal{N}_f$	number of offline fingerprints	

approximate displacement between two consecutive WiFi fingerprinting localization rounds. Then, a Monte Carlo Simulation (MCS) is designed to determine the transition probability by simulating the vehicle movement between the two WiFi fingerprinting localization rounds. The details of the MCS are described as follows.

First, we initialize a 2-dimensional uniform distribution over grid g_i and randomly get a set of samples $\chi_i=\{\mathbf{p}_{i1},\mathbf{p}_{i2},\ldots,\mathbf{p}_{i\mathcal{N}_s}\}$, where $\mathcal{N}_s$ is number of samples, and the q-th sample is a two-tuple $\mathbf{p}_{iq}=(x_{iq},y_{iq})$. Then, we update samples' position based on the vehicles' velocity $\mathbf{v}_{t-1:t}=\left\{v_{t-1:t}^{[1]},v_{t-1:t}^{[2]},\ldots,v_{t-1:t}^{[s]}\right\}$ and the heading direction $\mathbf{d}_{t-1:t}=\left\{d_{t-1:t}^{[1]},d_{t-1:t}^{[2]},\ldots,d_{t-1:t}^{[s]}\right\}$, where s is the number of sensor readings during the two consecutive fingerprinting localization rounds. Hence, the position of q-th sample after k-th update is computed by:

$$\left\{\begin{array}{l} x_{iq}^k=x_{iq}^{k-1}+\cos\left(d_{t-1:t}^{[k]}+\mathfrak{n}_d\right)\left(v_{t-1:t}^{[k]}+\mathfrak{n}_v\right)\Delta t\\ y_{iq}^k=y_{iq}^{k-1}+\sin\left(d_{t-1:t}^{[k]}+\mathfrak{n}_d\right)\left(v_{t-1:t}^{[k]}+\mathfrak{n}_v\right)\Delta t \end{array}\right. \tag{2}$$

where $\mathfrak{n}_v$ and $\mathfrak{n}_d$ follow the zero-mean Gaussian distribution, which represent the measurement noises of velocity and heading direction, respectively. Δt is the time interval between two consecutive sensor readings. After each updating, the map constraints are utilized to eliminate outlier samples (e.g., positions violate the map constraints). The details of map constraint generation and samples' elimination can be referred to [14]. In this way, the vehicle movement between two fingerprinting localization rounds can be simulated and the grids with fewer samples will have lower probability that the vehicle would transit to the corresponding state. When all the samples originated from g_i have been updated with the sequential velocity and direction readings, the transition probability is the number of samples at each grid over the total number of samples, which is computed by:

$$P\left(g_j\mid g_i\right)=\frac{\sum_{q=1}^{\mathcal{N}_s}\mathbb{I}_{\left(x_j^{(gb)}<x_{iq}<x_j^{(gu)}\right)\cup\left(y_j^{(gb)}<y_{iq}<y_j^{(gu)}\right)}}{\mathcal{N}_s} \tag{3}$$

where $(x_j^{(gb)},y_j^{(gb)})$ and $(x_j^{(gu)},y_j^{(gu)})$ are the coordinates of the j-th grid's left bottom corner and right upper corner, respectively.

3.2 Online Calibration and Evaluation

3.2.1 Calibration

With above formulated HMM, HOE further calibrates the WiFi fingerprinting localization based on the DR estimated displacement. Specifically, a forward probability is modelled to represent the probability of the vehicle on grid g_i when the current measurement sequence is $\{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_t\}$, which is denoted as $\boldsymbol{\alpha}_t=\left\{\alpha_t^i\mid 0<i<\mathcal{N}_g\right\}$, where $\alpha_t^i=P\left(\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_t,i_t=g_i\mid\lambda\right)$. As shown in Fig.2(a), the forward probability represents a discrete probability distribution of vehicles' location at round t. The forward probability can be transformed based on the transition probability, which is called the medium probability in our setting , and it is denoted by $\mathbf{O}_t=\left\{O_t^i\mid 1<i<\mathcal{N}_g\right\}$.

At the beginning, when there is no WiFi fingerprinting localization results, the forward probability $\boldsymbol{\alpha_t}$ is initialized by the initial state probability (i.e., $\boldsymbol{\alpha}_{t=0}=\pi$). When HOE receives a new WiFi fingerprinting localization result $\mathbf{z}_t$ and the corresponding displacement, the following three steps are conducted for online calibration.

Step 1: Calculate the transition probability $\mathbf{A}_t$ (at round t) for each grid based on Eq.(3) and update the medium probability $\mathbf{O}_t$ (at round t) based on the forward probability

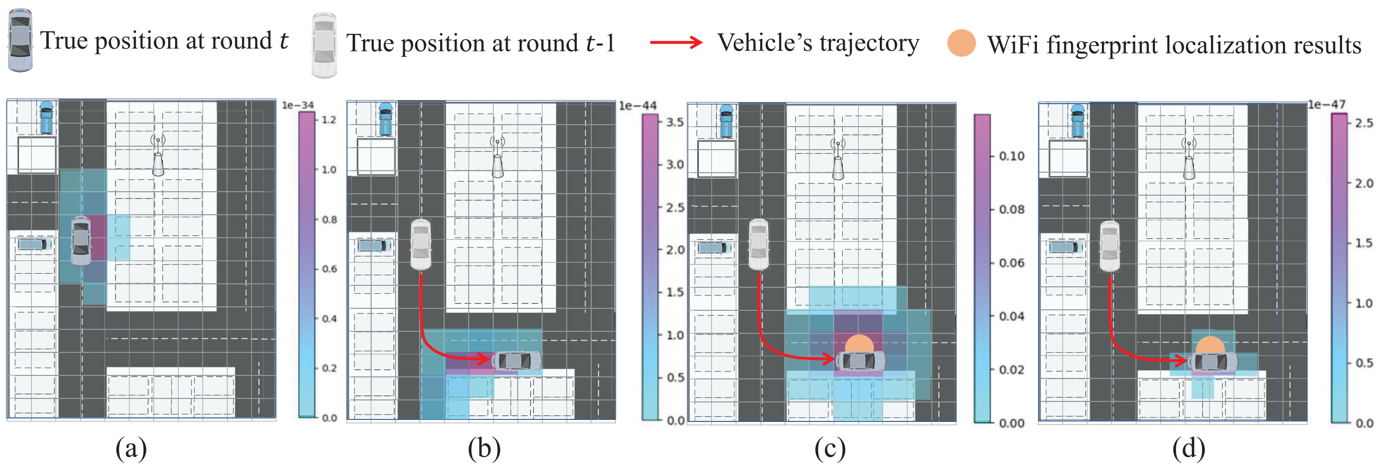


Fig. 2. The discrete probability distribution of the vehicle's position at different HOE steps. (a) The forward probability $\boldsymbol{\alpha}_{t-1}$ at round $t-1$; (b) the medium probability $\mathbf{O}_t$ at round t; (c) The emission probability $\mathbf{B}(\mathbf{z}_t)$ at round t; (d) The forward probability $\boldsymbol{\alpha}_t$ at round t.

$\boldsymbol{\alpha}_{t-1}$ (at round $t-1$) and the transition probability $\mathbf{A}_t$. The medium probability on i-th grid is computed by:

$$O_t^i = \sum_{j=1}^{\mathcal{N}_g} \alpha_{t-1}^j A_t^{ij} \tag{4}$$

With the medium probability of each grid, the $\mathbf{O}_t$ at round t can be derived. As shown in Fig.2(b), it forms a new distribution of vehicle's positions.

Step 2: Model the WiFi fingerprinting localization result $\mathbf{z}_t$ as the emission probability $\mathbf{B}\left(\mathbf{z}_t\right)$ according to Eq.(1). As shown in Fig.2(c), $\mathbf{B}\left(\mathbf{z}_t\right)$ is also a discrete probability distribution of vehicle's positions.

Step 3: Combine the two probability distributions $\mathbf{O}_t$ and $\mathbf{B}\left(\mathbf{z}_t\right)$ to form the forward probability at round t, which is computed by:

$$\alpha_t^i = O_t^i b_i\left(\mathbf{z}_t\right) \tag{5}$$

As shown in Fig.2(d), the forward probability at round t can better represent the vehicle's true location compared with the emission probability $\mathbf{B}\left(\mathbf{z}_t\right)$. Therefore, given the forward probability, the vehicle's position $\mathbf{p}^{(v)}$ is predicted by:

$$\mathbf{p}^{(v)} = \left(x_t^{(v)}, y_t^{(v)}\right) = \left(\sum_{i=1}^{\mathcal{N}_g} \frac{\alpha_t^i x_i^{(g)}}{\sum_{j=1}^{\mathcal{N}_g} \alpha_t^i}, \sum_{i=1}^{\mathcal{N}_g} \frac{\alpha_t^i y_i^{(g)}}{\sum_{j=1}^{\mathcal{N}_g} \alpha_t^i}\right) \tag{6}$$

where α_t^i is the forward probability of the i-th grid. In this way, the forward probability is normalized as the weight for each grid and the weighted average of all the grids is the final calibrated vehicle's location.

3.2.2 Evaluation

The design rationale of online evaluation is that the displacement inferred by DR should approximately connect the two fingerprinting localization results if the fingerprinting localization results were accurate. Recall that in **Step 1** of online calibration, the medium probability $\mathbf{O}_t$ represents the probability distribution of vehicle's positions after being updated by the DR estimated displacement (as shown in Fig.2(b)). Then, the WiFi fingerprinting localization $\mathbf{z}_t$ is modelled as the emission probability $\mathbf{B}\left(\mathbf{z}_t\right)$ (i.e., a discrete form of Gaussian distribution, as shown in Fig.2(c)). Therefore, we measure the inconsistency between these two distributions, which is used as the metric of WiFi fingerprinting localization accuracy. Specifically, the evaluation procedures are presented as follow.

First, we define a convergence criterion for forward probability $\mathbf{A}_t$ as follows:

$$\mathbb{C} = \sqrt{\left(x_t^{(v)} - x_{\mathcal{M}}^{(g)}\right)^2 + \left(y_t^{(v)} - y_{\mathcal{M}}^{(g)}\right)^2} \tag{7}$$

where $\mathcal{M}$ is the sequence number of grid that has the highest forward probability, which is computed by $\mathcal{M} = \operatorname{argmax}_i \alpha_t^i$. Recall that the forward probability is initialized by a uniform distribution during online calibration, the medium probability can not well represent the vehicle's location distribution when transformed from the initial forward probability. Therefore, when $\mathbb{C}$ is lower than a predefined threshold, HOE starts the online evaluation. When HOE receives the t-th fingerprinting localization result, we measure the inconsistency between the emission probability $\mathbf{B}\left(\mathbf{z}_t\right)$ and the medium probability $\mathbf{O}_t$ based on the Wasserstein distance, which can be seen as the ”minimum amount of work” required to transform the probability distribution $\mathbf{P}_1$ into $\mathbf{P}_2$, where the “work” is measured as the amount of distribution weight that must be moved, multiplied by the distance it has to be moved [15]. The Wasserstein distance is calculated as:

$$W_d\left(\mathbf{P}_1, \mathbf{P}_2\right) = \left(\inf_{J \in \mathcal{J}\left(\mathbf{P}_1, \mathbf{P}_2\right)} \int \|\mathbf{x} - \mathbf{y}\|^p d\mathcal{J}(\mathbf{x}, \mathbf{y})\right)^{1/p} \tag{8}$$

where $\mathcal{J}(\mathbf{P}_1, \mathbf{P}_2)$ denotes the all the joint distributions of $\mathcal{J}(\mathbf{x},\ \mathbf{y})$ that have margins $\mathbf{P}_1$ and $\mathbf{P}_2$.

A lower value of Waseerstein distance between $\mathbf{O}_t$ and $\mathbf{B}(\mathbf{z}_t)$ (i.e., $W_d\left(\mathbf{O}_t, \mathbf{B}(\mathbf{z}_t)\right)$) means that the two probability distributions are closer to each other, indicating a higher accuracy of the WiFi fingerprinting localization result. To make the evaluation more sensitive to the Wasserstein distance, a logarithm function is utilized to map the $W_d\left(\mathbf{O}_t, \mathbf{B}\left(\mathbf{z}_t\right)\right) \subset [0,1]$ to $[0,+\infty]$. Finally, a threshold $\mathcal{T}$ is set to identify whether the current WiFi fingerprinting localization result $\mathbf{z}_t$ is accurate or not, which is computed by:

$$e\left(\mathbf{z}_t\right) = \begin{cases} 1, & \log W_d\left(\mathbf{O}_t, \mathbf{b}\left(\mathbf{z}_t\right)\right) \geq \mathcal{T} \\ 0, & \log W_d\left(\mathbf{O}_t, \mathbf{b}\left(\mathbf{z}_t\right)\right) < \mathcal{T} \end{cases} \tag{9}$$

In our setting, the distance between WiFi fingerprinting localization results and the ground truth is less than the average WiFi fingerprint localization error, they are considered as the accurate ones. Then, the OTL is able to select the appropriate labels for current online RSSI samples, which forms the basis of online training for the WiFi fingerprinting localization model.

3.2.3 Computation Overhead

First, the time complexity of the forward algorithm for HMM is $O(TN^2)$, where N is the number of hidden states and T is the length of the sequence. As a variant of forward algorithm, HOE computes the vehicle location sequentially at each localization round. Therefore, the complexity for a single localization round is $O(\mathcal{N}_g^2)$, where $\mathcal{N}_g$ is the number of grids. In addition, note that the number of hidden states is relatively small in practice (i.e., in the order of hundred grids), and hence such a computation overhead is reasonable.

Second, the MCS procedure initializes a number of samples to simulate the displacement of the vehicle based on kinematic model, and repeats the procedures for each grid to determine the transition probability, which may cause extra overhead. In practice, this procedure only requires a small amount of samples (i.e., 30 samples) for each grid to realize the desired evaluation and localization performance, so the computation overhead is tolerable.

Finally, in Section V, we have conducted experiments to compare the running time of different algorithms for localization, and the results further verify that the computation overhead of the proposed algorithm would not undermine system scalability.

4 Online Transfer Learning Algorithm

In this section, we first propose an online transfer learning (OTL) algorithm, which incorporates the offline fingerprints and the online RSSI samples to enhance the adaptiveness of vehicle tracking in dynamic environments. Then, we derive the loss bounds of OTL with respect to offline fingerprints database and WiFi fingerprinting localization.

4.1 OTL Design

The OTL contains four schemes, including a WiFi fingerprinting localization scheme, a label conversion scheme, a weight allocation scheme, and an instance-based transferring scheme. Their primary functions are summarized as follows. The WiFi fingerprinting localization scheme consists of two classification models, namely, an batch model trained by the offline fingerprint database, and an online model trained by online RSSI samples. The label conversion scheme is adopted to construct a candidate label set for the current online RSSI sample to train the online classification model. Then, the weight allocation scheme is designed to calculate the loss for each classification model based on the candidate label set and adjust the model weights accordingly. Finally, the instance-based transferring scheme is designed to transfer offline fingerprints that are more suitable for current signal features to retrain the batch model. Details are elaborated below.

WiFi fingerprinting localization scheme: it adopts the support vector machine (SVM) [16] as the batch classification model h_1 and adopts the Logistic Regression [17] with Stochastic Gradient Descent (SGD) [18] as the online classification model h_2. The batch model is trained based on the offline fingerprint database and the online model is trained using a few randomly selected offline fingerprints as a warm start. The offline fingerprint database is denoted by $R_s = \{(\mathbf{r}_i, l_i) \mid i \in [1, \mathcal{N}_f]\}$, where $\mathbf{r}_i$ is a d-dimensional RSSI sample (d is APs' number), $\mathcal{N}_f$ is the number of total fingerprints and l_i is the ID of the corresponding RP (i.e., the label). Given $\mathcal{N}_r$ RPs, the coordinate of the RP l_i is denoted by $\mathbf{p}_{l_i}^{(rp)} = \left(x_{l_i}^{(rp)}, y_{l_i}^{(rp)}\right)$, and hence the classification model has $\mathcal{N}_r$ classes.

When receiving an online RSSI sample $\bar{\mathbf{r}}_t$ at round t, the classification models predicted the confidence for each RP, which is denoted by $h\left(\bar{\mathbf{r}}_t\right) = \{P\left(l_j \mid \bar{\mathbf{r}}_t\right) \mid 1 < j < \mathcal{N}_r\}$. K RPs with the largest confidence are considered as the predicted RPs of the classification model o, which is denoted by $\check{\mathbf{L}}_t^o$. The localization results of the classification model are the weighted average of the predicted RPs' coordinates based on their confidence. Finally, two weights are introduced in the proposed fingerprinting localization scheme to integrate the localization results of two classification models, which is denoted by θ_n^1 and θ_n^2, respectively, where $\theta_n^{o=1,2} \in [0, 1]$.

Label conversion scheme: If the t-th fingerprinting localization results is evaluated as accurate, the predicted RPs of the two models (i.e., $\check{\mathbf{L}}_t^1, \check{\mathbf{L}}_t^2$) are selected to construct the candidate label set for online RSSI samples. Since the two classification models are combined to estimated the WiFi fingerprinting localization results, the predicted RPs of the two models may be closer to the vehicle's true location when the fingerprinting localization results are evaluated as accurate. In order to select the effective labels from the predicted RPs, a distance threshold $\mathcal{T}_D$ is introduced to identify whether the predicted RPs are close to the fingerprinting localization results. For those RPs whose distance to the fingerprint localization result is lower than $\mathcal{T}_D$ are selected as the candidate labels. In addition, if the fingerprinting localization results are evaluated as inaccurate, the RPs with the closest distance to the calibrated vehicle's position $\mathbf{p}^{(v)}$ can be also considered as the labels when the amount of accurate fingerprinting localization results are insufficient.

Weight allocation scheme: the losses of the two models are calculated based on the candidate label set, which are further utilized to adjust the weights of the two models. At the beginning, the two weights are assigned with the same value (i.e., $\theta_{n=1}^1 = \theta_{n=1}^2 = 0.5$). Then, the Hedge algorithm proposed in [19] is adopted to adjust the weight of the two models based on their losses, so that the model with less loss will be assigned with a higher weight. In this way, the weight allocation scheme realizes cooperation of the two classification models.

Instance-based transferring scheme: we assign a set of weights for the offline fingerprints to realize the transferring and resampling, which is denoted by $\mathbf{w}_m = \{w_m^i \mid 1 < i < \mathcal{N}_f\}$ in the m-th instance-based transferring, where $w_m^i \in [0,1]$. All the weights are equal at the beginning (i.e., $w_{m=1}^i = 1/\mathcal{N}_f$). Then, each offline fingerprint is tested using the online model. If the fingerprints' location are close to the predicted results of the online model, these fingerprints can better capture the signal features of current wireless environment. Accordingly, the Hedge algorithm [19] is adopted to reduce the weights for offline fingerprints that are distant to the predicted results. Based on the updated weight of each fingerprint, we can resample the offline fingerprint database to further train the batch model. In addition, two discounting parameters for penalizing the classification models and fingerprints are introduced, denoted by B and β, respectively.

The detailed procedures of OTL are presented as follows, which are shown in the pseudo-code in **Algorithm 1**.

(1) Initialization (line 1): The two weights $\theta_{n=1}^1$ and $\theta_{n=1}^2$ for the classification models are initialized as $1/2$, and the weight $w_{m=1}^i \quad (i \in [1, \mathcal{N}_f])$ for each fingerprint is initialized as $1/\mathcal{N}_f$. Furthermore, the initial training for the two classification models h_1 and h_2 are conducted based on the offline fingerprint database R_s.

(2) WiFi fingerprinting localization (lines 3-6): When receiving an online RSSI sample $\bar{\mathbf{r}}_t$ at round t, the two classification models h_1 and h_2 predict the probabilistic results based on the online RSSI sample. For the results of model h_o, where $o \in \{1,2\}$, we select k RPs with the largest probabilities as the predicted RP $\check{\mathbf{L}}_t^o = \{l_1^o, l_2^o, \ldots, l_k^o\}$ and normalize their probabilities as weights, and the weighted average of these RPs' positions is considered as the final localization result of h_o, which is computed by:

$$\tilde{\mathbf{z}}_t^o = \left(\sum_{i=1}^{\left|\check{\mathbf{L}}_t^o\right|} \frac{P\left(l_i^o \mid \bar{\mathbf{r}}_t\right) x_{l_i^o}^{(rp)}}{\sum_{j=1}^{\left|\check{\mathbf{L}}_t^o\right|} P\left(l_j^o \mid \bar{\mathbf{r}}_t\right)}, \sum_{i=1}^{\left|\check{\mathbf{L}}_t^o\right|} \frac{P\left(l_i^o \mid \bar{\mathbf{r}}_t\right) y_{l_i^o}^{(rp)}}{\sum_{j=1}^{\left|\check{\mathbf{L}}_t^o\right|} P\left(l_j^o \mid \bar{\mathbf{r}}_t\right)} \right) \tag{10}$$

Next, we combine the localization results of the two models based on their corresponding weights. Suppose the weight allocation scheme has been conducted n times, the combined localization result is computed by

$$\mathbf{z}_t = \left(x_t^{(f)}, y_t^{(f)}\right) = \epsilon_n^1 \tilde{\mathbf{z}}_t^1 + \epsilon_n^2 \tilde{\mathbf{z}}_t^2 \tag{11}$$

where $\epsilon_n^o = \theta_n^o / \left(\sum_{o=1}^2 \theta_n^o\right)$ is the normalized weight of two models.

(3) Label conversion (lines 7-9): If $\mathbf{z}_t$ was evaluated as accurate (i.e., $e\left(\mathbf{z}_t\right) = 1$), then, get the union set of RPs predicted by the two classification models, which is denoted by $\check{\mathbf{L}}_t = \check{\mathbf{L}}_t^1 \cup \check{\mathbf{L}}_t^2$. Further, the candidate labels are selected from RPs in the union set based on their distances to the localization result $\mathbf{z}_t$, and the candidate label set is obtained by $\mathbf{L}_t^c = \left\{l_i \in \check{\mathbf{L}}_t \mid \left\|\mathbf{z}_t - \mathbf{p}_{l_i}^{(rp)}\right\|_2 < \mathcal{T}_D\right\}$, where $\mathcal{T}_D$ is a predetermined distance threshold. Finally, the online model can be trained with the candidate label set $\mathbf{L}_t^c$ with corresponding online RSSI sample $\bar{\mathbf{r}}_t$. In practice, a batch of online RSSI samples and their candidate label sets are cached, and then the online model is trained in a mini-batch manner. On this basis, the weight allocation scheme is conducted for each pair of the online sample and the corresponding candidate label set $(\bar{\mathbf{r}}_t, \mathbf{L}_t^c)$ in the batch.

(4) Weight allocation (lines 16-18): For each label $l_i^c \in \mathbf{L}_t^c$, compute the loss for the two classification models by $\mathcal{L}_n^o = \left[\operatorname{argmax}_{l_j} h_o\left(\bar{\mathbf{r}}_t\right) \neq l_i^c\right]$, where $[x] = 1$ if x was true and $[x] = 0$ otherwise. The weight parameter for the n-th weight allocation of each model is updated by:

$$\theta_{n+1}^o = \theta_n^o B^{\mathcal{L}_n^o} \tag{12}$$

where $o = 1, 2$.

(5) Instance-based transferring (lines 21-25): First, input the offline fingerprints into the online model for prediction. Then, calculate the loss for each offline fingerprint based on its distance to the label predicted by the online model. The label predicted by online model is $l_a = \operatorname{argmax}_{l_j} h_2\left(\mathbf{r}_i\right)$, the loss of i-th offline fingerprints is $\ell_m^i = \left[\left\|\mathbf{p}_{l_a}^{(rp)} - \mathbf{p}_{l_i}^{(rp)}\right\|_2 > \mathcal{T}_S\right]$, where $\mathcal{T}_S$ is a predefined distance threshold. On this basis, the weight of i-th fingerprint after m-th instance-based transferring is computed as follows:

$$w_{m+1}^i = w_m^i \beta^{\ell_m^i} \tag{13}$$

Finally, the offline fingerprints are resampled based on their weights and the batch classification model is retrained with the resampled offline fingerprints. Note that the instance-based transferring scheme is conducted periodically during the online phase.

Algorithm 1 Online Transfer Learning Algorithm

```
Input: The initial labelled WiFi fingerprint database $R_s$
       A batch model $h_1$ and an online model $h_2$
 1: Initialize $n = 1$, $m = 1$, $\theta_n^1 = 1/2$, $\theta_{n=1}^2 = 1/2$, $w_m^i = 1/\mathcal{N}_f$ where $i \in [1, \mathcal{N}_f]$, a cache set $S = \{\}$
 2: for $t = 1, 2, .., T$ do
 3:   receive an online RSSI measurement $\bar{\mathbf{r}}_t$
 4:   $\check{\mathbf{L}}_t^o = h_o(\bar{\mathbf{r}}_t)$, where $o = 1, 2$
 5:   compute $\tilde{\mathbf{z}}_t^o$ based on Eq.(10), where $o = 1, 2$
 6:   compute $\mathbf{z}_t$ based on Eq.(11)
 7:   if $e(\mathbf{z}_t) == 1$ then
 8:     $\check{\mathbf{L}}_t = \check{\mathbf{L}}_t^1 \cup \check{\mathbf{L}}_t^2$
 9:     $\mathbf{L}_t^c = \left\{l_i \in \check{\mathbf{L}}_t \mid \left\|\mathbf{z}_t - \mathbf{p}_{l_i}^{(rp)}\right\|_2 < \mathcal{T}_D\right\}$
10:     $S = S \cup \{(\bar{\mathbf{r}}_t, \mathbf{L}_t^c)\}$
11:   end if
12:   if $t$ % interval==0 then
13:     train $h_2$ with online sample cached in $S$
14:     for $(\bar{\mathbf{r}}_t, \mathbf{L}_t^c) \in S$ do
15:       for $l_i^c \in \mathbf{L}_t^c$ do
16:         $\mathcal{L}_n^o = \left[\arg\max_{l_j} h_o\left(\bar{\mathbf{r}}_t\right) \neq l_i^c\right]$, where $o = 1, 2$
17:         $\theta_{n+1}^o = \theta_n^o B^{\mathcal{L}_n^o}$
18:         $n = n + 1$
19:       end for
20:     end for
21:     compute the loss vector $\boldsymbol{\ell}_m = (\ell_m^1, \ell_m^2, ...\ell_m^{\mathcal{N}_f})$
22:     update weights for each offline fingerprints $w_{m+1}^i = w_m^i \beta^{\ell_m^i}$
23:     set $\boldsymbol{\rho}_m = \mathbf{w}_m / \left(\sum_{i=1}^{\mathcal{N}_f} w_m^i\right)$
24:     Retrain $h_1$ based on the new distribution $\boldsymbol{\rho}_m$ over fingerprints
25:     $m = m + 1$, $S = \{\}$
26:   end if
27: end for
```

4.2 Loss Bound Analysis

This part derives the loss bound of the offline fingerprint database constructed via the instance-based transferring scheme, as well as the loss bound of the WiFi fingerprint localization when combing the two classification models via the weight allocation scheme.

First, we define the loss of the offline fingerprint database through M rounds of instance-based transferring as follows.

$$L_{(R_s)} = \sum_{m=1}^{M} \boldsymbol{\rho}_m \boldsymbol{\ell}_m = \sum_{m=1}^{M} \sum_{i=1}^{\mathcal{N}_f} \rho_m^i \ell_m^i \tag{14}$$

where $\rho_m^i = w_m^i / \left(\sum_{j=1}^{\mathcal{N}_f} w_m^j \right), (i = 1, 2, ...\mathcal{N}_f)$ is the normalized weights of each fingerprints at m-th instance-based transferring and $\boldsymbol{\rho}_m = \left\{ \rho_m^1, \rho_m^2, ..., \rho_m^{\mathcal{N}_f} \right\}$. On the other hand, the loss suffers from the i-th offline fingerprint through M rounds of transferring is denoted by $L_{(\mathbf{r}_i)} = \sum_{m=1}^{M} \ell_m^i$. Then, we have following theorem.

Theorem 1. With M rounds of instance-based transferring, the total loss of the entire offline fingerprints is bounded by:

$$L_{(R_s)} \leq \min_i L_{(\mathbf{r}_i)} + \sqrt{2M \ln \mathcal{N}_f} + \ln \mathcal{N}_f \tag{15}$$

Theorem 1 indicates that the total loss of the entire offline fingerprint database through M rounds of transferring suffered from the online model $L_{(R_s)}$ is bounded by $\sqrt{2M \ln \mathcal{N}_f} + \ln \mathcal{N}_f$ compared with the minimum loss of a single offline fingerprint (i.e., $\min_i L_{(\mathbf{r}_i)}$), which demonstrates the effectiveness of the instance-based transferring scheme on selecting more suitable fingerprints for current signal features.

The proof of Theorem 1 follows the idea presented in [19]. First, we prove two lemmas as below.

Lemma 1. After conducting M rounds of instance-based transferring, the loss of offline fingerprint database is bounded by

$$L_{(R_s)} = \sum_{m=1}^{M} \sum_{i=1}^{\mathcal{N}_f} \rho_m^i \ell_m^i \leq \frac{\ln \left(\sum_{i=1}^{\mathcal{N}_f} w_{M+1}^i \right)}{-(1-\beta)} \tag{16}$$

Proof By a convexity argument, for $\alpha \geq 0$ and $r \in [0, 1]$, we have

$$\alpha^r \leq 1 - (1-\alpha) r$$

Therefore:

$$\begin{aligned} \sum_{i=1}^{\mathcal{N}_f} w_{m+1}^i &= \sum_{i=1}^{\mathcal{N}_f} w_m^i \beta^{\ell_m^i} \\ &\leq \sum_{i=1}^{\mathcal{N}_f} w_m^i (1 - (1-\beta) \ell_m^i \\ &= (\sum_{i=1}^{\mathcal{N}_f} w_m^i)(1 - (1-\beta) \boldsymbol{\rho}_m \boldsymbol{\ell}_m) \end{aligned} \tag{17}$$

Since $\ln(1-x) \leq -x$ for $x \in [0, 1]$, we have

$$\ln \left(\sum_{i=1}^{\mathcal{N}_f} w_{m+1}^i \right) \leq \ln \left(\sum_{i=1}^{\mathcal{N}_f} w_m^i \right) - (1-\beta) \boldsymbol{\rho}_m \boldsymbol{\ell}_m \tag{18}$$

Recursively applying for $m = 1, 2, ...M$ yields

$$\begin{aligned} \ln \left(\sum_{i=1}^{\mathcal{N}_f} w_{M+1}^i \right) &\leq \ln \left(\sum_{i=1}^{\mathcal{N}_f} w_1^i \right) - (1-\beta) \sum_{m=1}^{M} \boldsymbol{\rho}_m \boldsymbol{\ell}_m \\ &\leq -(1-\beta) \sum_{m=1}^{M} \boldsymbol{\rho}_m \boldsymbol{\ell}_m \\ &= -(1-\beta) L_{(R_s)} \end{aligned} \tag{19}$$

Lemma 1 is proved.

Lemma 1 gives an upper bound of the loss of the offline fingerprint database based on the weights assigned for each fingerprints.

Lemma 2. Suppose $0 \leq L \leq \tilde{L}$ and $0 \leq R \leq \tilde{R}$. Let $\beta = 1/(1 + \sqrt{2\tilde{R}/\tilde{L}})$, Then

$$\frac{-L ln(\beta) + R}{1-\beta} \leq L + \sqrt{2\tilde{L}\tilde{R}} + R \tag{20}$$

Proof since $-\ln \beta \leq (1-\beta^2)/(2\beta)$ when $\beta \in (0, 1]$, we have

$$\begin{aligned} \frac{-L \ln \beta + R}{1-\beta} &\leq \frac{L \frac{1-\beta^2}{2\beta} + R}{1-\beta} \\ &= L + R + \sqrt{\frac{\tilde{R} L^2}{2\tilde{L}}} + \sqrt{\frac{\tilde{L} R^2}{2\tilde{R}}} \\ &\leq L + \sqrt{2\tilde{L}\tilde{R}} + R \end{aligned} \tag{21}$$

Lemma 2 reveals how to choose the discounting factor.

By setting $\beta = 1/\left(1 + \sqrt{2 \ln \mathcal{N}_f / M}\right)$, Theorem 1 is proved as follows.

Proof for any nonempty set $C \subseteq \{1, ..., \mathcal{N}_f\}$, we have

$$\begin{aligned} \sum_{i=1}^{\mathcal{N}_f} w_{M+1}^i &\geq \sum_{i \in C} w_{M+1}^i \\ &= \sum_{i \in C} w_1^i \beta^{L_{(\mathbf{r}_i)}} \\ &\geq \beta^{\max_{i \in C} L_{(\mathbf{r}_i)}} \sum_{i \in C} w_i^1 \end{aligned} \tag{22}$$

Then With Lemma 1, we have

$$L_{(R_s)} \leq \frac{-\ln(\sum_{i \in C} w_1^i) - \max_{i \in C} L_{(\mathbf{r}_i)} \ln \beta}{1-\beta} \tag{23}$$

When setting a special case $C = \{i\}$, and since $\mathbf{w_1}$ is initialized as a uniform distribution (i.e., $w_1^i = \frac{1}{\mathcal{N}_f}$), the bound becomes

$$L_{(R_s)} \leq \frac{-\min_{1 \leq i \leq \mathcal{N}_f} L_{(\mathbf{r}_i)} ln(\beta) + \ln \mathcal{N}_f}{1-\beta} \tag{24}$$

An upper bound on cumulative loss of the best offline fingerprint is $L_{(\mathbf{r}_i)} = M$ and there are total $\mathcal{N}_f$ offline fingerprints. Therefore, we can set $\tilde{L} = M$ and $\tilde{R} = \ln \mathcal{N}_f$. Based on Lemma 2 and the chosen $\beta = 1/\left(1 + \sqrt{2 \ln \mathcal{N}_f / M}\right)$, we can convert the Eq.(24) into

$$L_{(R_s)} \leq \min_i L_{(\mathbf{r}_i)} + \sqrt{2M \ln \mathcal{N}_f} + \ln \mathcal{N}_f \tag{25}$$

□ Theorem 1 is proved.

Next, we analyze the loss bound of WiFi fingerprinting localization, and we have the following theorem.

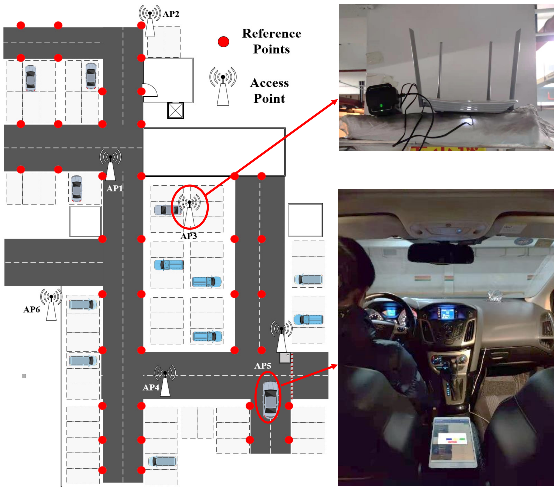


Fig. 3. Experiment deployment

Theorem 2. With N rounds of weight allocation, the loss of the WiFi fingerprinting localization Φ is bounded by:

$$\Phi \le \min_{i\in\{1,2\}} \Phi_i + \sqrt{2N\ln 2} + \ln 2 \tag{26}$$

where $\Phi = \sum_{n=1}^{N}\sum_{o=1}^{2} \epsilon_n^o \mathcal{L}_n^o$ and $\epsilon_n^o = \theta_n^o / \sum_{j=1}^{2} \theta_n^j$ is the normalized weight for the classification model o. Note that $\Phi_i = \sum_{n=1}^{N} \mathcal{L}_n^i$, which is the total loss suffered from the classification model h_i.

Theorem 2 indicates that the localization performance of the WiFi fingerprinting localization scheme is close to the best classification model during the whole online trials, which verifies that weight allocation scheme is able to assign probable weights to the classification models. The proof is given below.

Proof Recall that Lemma 1 gives an upper bound for the loss of $\mathcal{N}_f$ fingerprints. For the loss of two models, by setting $\mathcal{N}_f = 2$, we have

$$\Phi = \sum_{n=1}^{N}\sum_{o=1}^{2} \epsilon_n^o \mathcal{L}_n^o \le \frac{-\ln\left(\sum_{o=1}^{2} \epsilon_{N+1}^o\right)}{1-B} \tag{27}$$

Since $\sum_{o=1}^{2} \epsilon_{N+1}^o \ge \epsilon_{N+1}^i = \epsilon_1^i B^{\sum_{n=1}^{N} \mathcal{L}_n^i} = \epsilon_1^i B^{\Phi_i}$, where $i \in \{1,2\}$, we have

$$\Phi \le \frac{-ln(B)\min_{i\in\{1,2\}} \Phi_i + \ln 2}{1-B} \tag{28}$$

Then, based on Lemma 2, by setting the weight discounting parameter B used in the weight allocation scheme as $1/(1+\sqrt{2\ln 2/N})$, we have

$$\Phi \le \min_{i\in\{1,2\}} \Phi_i + \sqrt{2N\ln 2} + \ln 2 \tag{29}$$

□ Theorem 2 is proved.

5 Performance Evaluation

5.1 Experimental Environments and Settings

The experiments were conducted in a typical underground parking area. The layout of the experiment area and the corresponding AP deployment are shown in Fig.3. Specifically, the area covers around 3500 m^2 and contains 63 parking slots. We deployed 5 APs and 40 RPs in the area and the corresponding positions are shown in Fig.3. During the offline training, 40 fingerprints were collected at each RP and the sample interval between two fingerprints is 5 seconds. An Android-based APP is developed and installed on the mobile device, which is placed in the vehicle to collect the inertial sensors' readings at the frequency of 25Hz, and to collect WiFi RSSIs at the frequency of 0.5Hz.

TABLE 2
Confusion Matrix

Evaluation Ground Truth	Accurate	Inaccurate	**Recall:** **37/(37+3)=0.925**
Accurate	37	3	
Inaccurate	8	16	
Precision: 37/(37+8)=0.82			

We have driven more than 200 rounds in the experiment area and the total driving distance is over 20 km. 90% of the data are used to analyze the RSSI changing characteristics and train the parameters for HOE and OTL, and 10% of data are used for testing. To obtain the ground truth, we took the video during the testing, and the landmarks including the parking slots, pillars, gutterway, were adopted to infer vehicle's true positions.

To quantitatively evaluate the algorithm performance, the following metrics are adopted.

- Recall (a.k.a. True positive rate): denote the number of accurate fingerprinting localization results that are correctly and incorrectly evaluated by HOE as N_{tp} and N_{fn}, respectively, then the Recall is computed by:
$$Recall = N_{tp}/(N_{tp} + N_{fn})$$
- Precision: denote the number of inaccurate fingerprinting localization results that are incorrectly evaluated by HOE as N_{fp}, then the Precision is computed by:
$$Precision = N_{tp}/(N_{tp} + N_{fp})$$
- False positive rate (FPR): denote the number of inaccurate fingerprinting localization results that are correctly evaluated by HOE as N_{tn}, then the FPR is computed by:
$$FPR = N_{fp}/(N_{tn} + N_{fp})$$
- Average localization error: it is defined as the average Euclidean distance between each estimated vehicle coordinate and its true coordinate.
- Cumulative Distribution Function (CDF) of localization error: it is the function that $y = P(X <= x)$, where X is the localization error.

5.2 Experimental Results

5.2.1 *The effectiveness of HOE*

Recall that HOE is designed to evaluate the WiFi fingerprinting localization results and identify whether they are accurate. In order to validate the effectiveness of HOE, we first

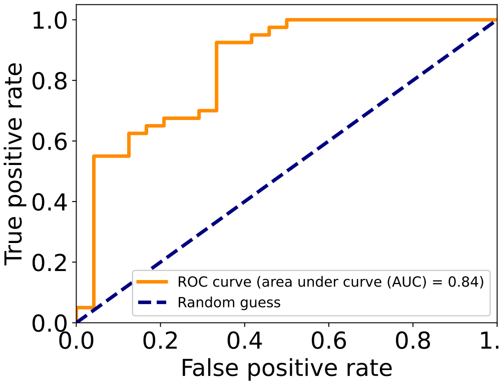


Fig. 4. ROC curve of HOE

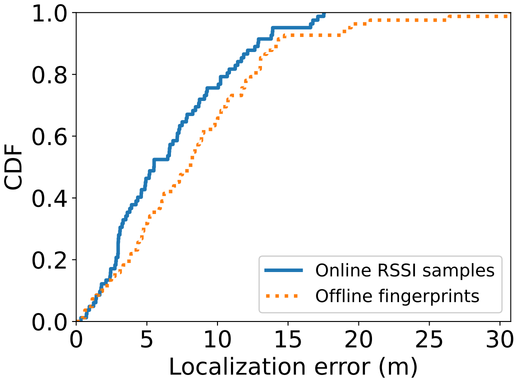


Fig. 5. The effectiveness of label conversion in OTL

conduct an experiment as follows. The online RSSI samples are utilized to predict the localization results based on the batch classification model. Then, the distance between the localization result and the vehicle's true location is adopted for deriving the ground truth. Specifically, 64 online RSSI samples were evaluated in the experiment, and 40 of the localization results were accurate. On this basis, the same set of WiFi fingerprinting localization results were evaluated by HOE, and the output was compared with the ground truth.

The confusion matrix is adopted to exhibit the performance of HOE. As shown in Table 2, there are in total 40 accurate WiFi fingerprinting localization results, and the HOE identified 37 of them. So, the recall is 0.925. On the other hand, 8 inaccurate fingerprinting localization results were identified as accurate, which gives the precision of 0.82. In addition, Fig.4 exhibits the Receiver Operating Characteristic (ROC) curve of HOE. The corresponding Area Under Cure (AUC) provides aggregated measure of performance across all possible decision thresholds. A higher value of AUC indicates better performance of HOE on identifying the accuracy of the WiFi fingerprinting localization results. As shown in Fig.4, the HOE achieves 0.84 of AUC, which demonstrates its effectiveness on evaluating WiFi fingerprinting localization results.

5.2.2 The effectiveness of OTL

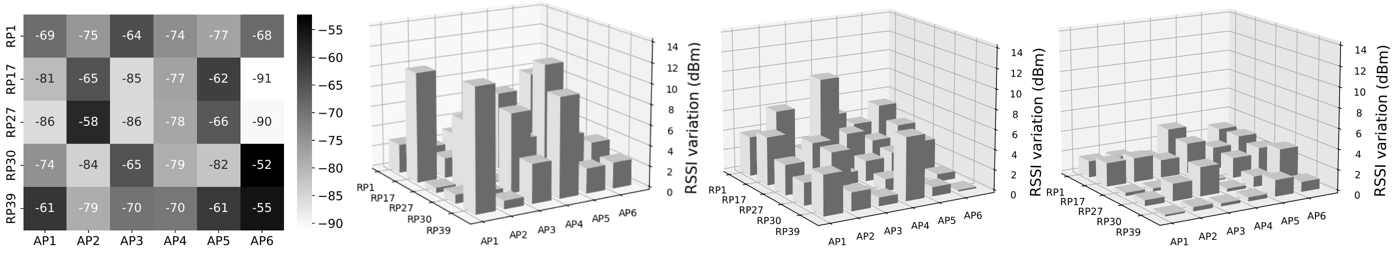
Recall that according to OTL, the online RSSI samples which are evaluated as accurate will be assigned with candidate labels. Then, the online RSSI samples and their candidate labels are ensembled to construct the online training set. So, we first evaluate the effectiveness of OTL in terms of labeling online RSSI samples. To do so, we compare the localization accuracy by using online training set and the offline fingerprint database. The K-Nearest-Neighbors (KNN) localization algorithm is adopted for both datasets. Fig.5 shows the CDF of the localization error. As shown, the online training set achieves higher localization accuracy than the offline fingerprint database, which clearly demonstrates the effectiveness of label conversion in OTL. More importantly, it verifies that the online RSSI samples can better reflect the current signal characteristics, and hence can be utilized to continuously update the classification model in dynamic wireless environments.

Next, we validate the effectiveness of the weight allocation and the instance-based transferring schemes under different environments. Specifically, three online training datasets were collected in the driving vehicle on different dates (i.e., Jan. 8th, 10th and 11th of 2021) for training OTL. In contrast, the testing dataset was collected on Jan. 11th. To reveal dynamic RSSI features over time, we calculate the average RSSI value of each AP at 5 RPs. Fig. 6(a) shows the absolute values of RSSI distribution on Jan. 11th, 2021, which reflects the signal features of the testing dataset. On this basis, Figs.6(b), (c) and (d) show the differences of RSSI distribution compared with the training datasets collected on Jan. 8th, 10th and 11th, 2021, respectively. As observed, the RSSI distribution difference is relatively smaller for the training dataset which was collected on the same date with the testing dataset, while the difference is getting larger when the collection date of training dataset was farther apart from the date of collecting testing dataset.

The convergence of average localization error of OTL under different environments are shown in Fig.7. As noted, the OTL can achieve similar performance under different environment dynamics. The average localization error of the online model decreases rapidly by training with the batch of online RSSI samples at each round. Also, the average localization error of the batch model decreases with periodical retraining, which demonstrates the effectiveness of the instance-based transferring scheme. On the other hand, with several rounds of training, the localization error of WiFi fingerprinting localization scheme is very close to the classification model that performs better, which demonstrates that the weight allocation scheme can assign proper weights to the two models.

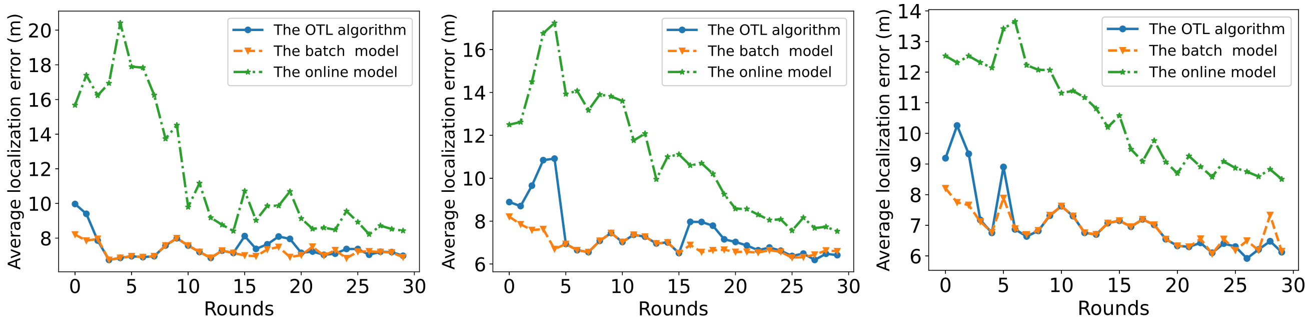
In addition, we compare OTL with representative online learning methods including Online SVM with SGD solver, Perceptron algorithm [20], Passive Aggressive Algorithm [21], and with two state-of-the-art online transfer learning algorithms, including HomOTL [22] and HomOTLMS [23]. These methods are trained with the same online training set as OTL, and their predicted labels for RSSI samples in the testing set are considered as the localization results.

The performance of different algorithms are shown in Fig.8 and the average localization errors are summarized in Table 3. As shown, compared with existing online learning methods, OTL achieves the best performance on minimizing the localization error, which further validates its advantage on incorporating offline fingerprints and online RSSI sam-



(a) (b) (c) (d)

Fig. 6. Variation of RSSI on different dates. (a) RSSI distribution of testing dataset collected on Jan. 11th, 2021. (b), (c) and (d) Difference of RSSI distribution between the testing dataset and the training dataset, which are collected on Jan. 8th, Jan. 10th, Jan. 11th of 2021, respectively.



(a) (b) (c)

Fig. 7. OTL performance evaluation with variation of RSSI feature distributions. (a), (b) and (c) OTL performance with the online training dataset collected on Jan. 8th, 10th and 11th of 2021, respectively.

TABLE 3
Fingerprinting localization performance of different online learning methods

	OTL	Online SVM	PA-1	Perceptron	HomOTL	HomOTLMS
Average localization error (m)	5.93	8.34	11.31	12.88	6.70	6.85

TABLE 4
Overall localization performance of different algorithms

	ParkLoc	VeTrack	VeTorch	AcMu	HOE+OTL
Average localization error (m)	7.08	6.39	10.84	8.16	4.54
Standard deviation of localization error (m)	5.28	5.86	8.87	5.06	2.80
Performance improvement	35.8%	28.9%	58.1%	44.3%	–

ples. Meanwhile, OTL outperforms the two online transfer learning algorithms. This is mainly because the designed instance-based transferring scheme can reconstruct the distribution of fingerprint to better accommodate the changes of wireless signal features. In contrast, HomOTL and HomOTLMS simply combine different models trained by the online samples and the offline fingerprints, without reusing the offline fingerprints.

5.2.3 Overall tracking evaluation

Finally, we evaluate the overall tracking performance of the online learning framework (i.e., HOE+OTL). Four competitive algorithms, namely, AcMu [10], VeTorch [24], ParkLoc [25] and VeTrack [26], are implemented for comparison. Details of these four algorithms can be referred to Section VI. For the comparison purpose, the WiFi fingerprinting localization results are considered as the landmark in VeTrack. The CDF of loccalization error of different algorithms is shown in Fig.9 and the corresponding statistics are summarized in Table 4. As shown, HOE+OTL achieves the best localization accuracy with mean errors of 4.54m, which improves the accuracy by 28.9%, 35.8%, 44.3% and 58.1% compared with VeTrack, ParkLoc, AcMu and VeTorch, respectively. Further, we choose Vetrack, which has the best tracking performance among comparative algorithms, to construct the whole vehicle trajectory. The tracking performance of the online learning framework and the VeTrack are shown in Fig.10. As demonstrated, the proposed online learning framework is closer to the ground truth.

Finally, we compare the running time of different al-

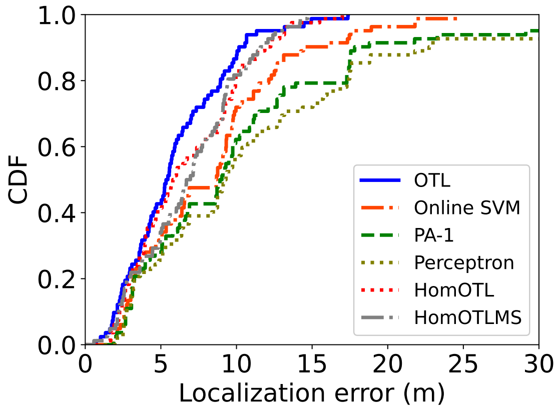


Fig. 8. CDF of localization error of different online learning methods

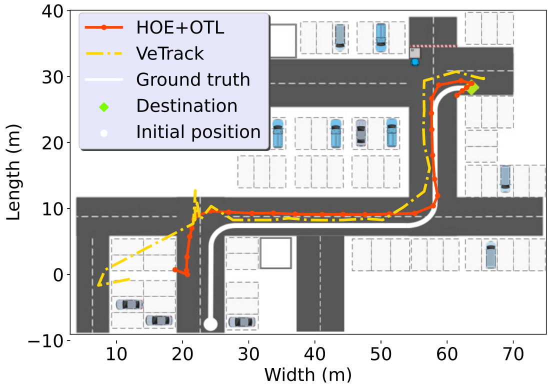


Fig. 10. Comparison of overall tracking performance

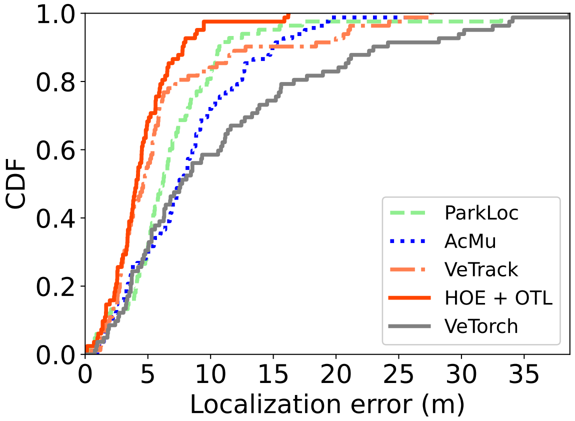


Fig. 9. CDF of localization error of different algorithms

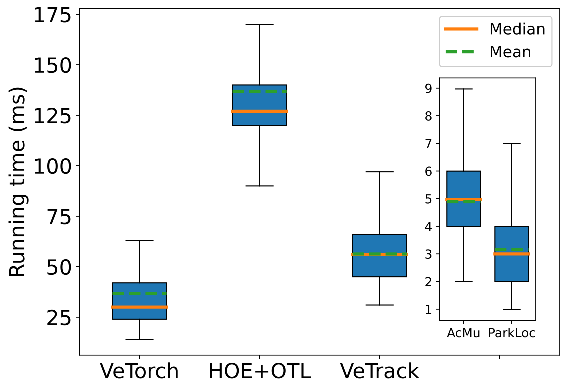


Fig. 11. Boxplot of running time of different algorithms

gorithms. The primary hardware configuration includes 3 GHz CPU and 16G memory, and the running time statistics of different algorithms are shown in Fig.11. The boxplot describes the minimum, the maximum (i.e., the top and bottom lines), the first quartile, the third quartile (i.e., the top and bottom boundaries of the box), the median and the mean of the running time. As shown, although the average running time of HOE+OTL is higher than that of other algorithms, it remains in the order of 100ms, which is feasible for vehicle tracking. Moreover, note that with an increasing of server computation capacities, the running time overhead of HOE+OTL could be negligible in practice.

6 Related Work

Numerous WiFi fingerprint-based methods have been proposed for indoor localization. As one of the pioneering schemes, RADAR [27] adopted empirical fingerprints and KNN to estimate the location. M. Youssef et al. [28] developed a localization system called Horus, which considers the probability distribution of RSSI features and leverages the maximum likelihood to enhance the localization performance. C. OWN et al. [29] designed a fusion learning method with the dual WiFi bands. It leverages SVM to distinguish the NLOS and LOS environments and employs the capsule network to derive users' positions. H. Zhang et al. [9] explored the dual-band signal for localization, which leverages stacked denoising autoencoder to efficiently extract RSSI features and alleviate the influence of RSSI noisy. S. Kumar et al. [30] leveraged the compartmental model to formulate the RSSI of different sensor networks in NLOS conditions. Then, a low-complexity mini-batch Singular Value Decomposition (SVD) method was proposed for target detection and localization. M. Zhou et al. [31] leveraged the integrated statistical test to examine the normality of WiFi signal distributions and the contribution of each AP to overall localization accuracy, which is further adopted to model the weight of RPs. M. Z. Win et al. [32] explored various network operation strategies including node prioritization, node activation and node deployment to improve the localization accuracy and prolong the network lifetime. J. Choi et al. [33] aimed to adaptively adjust the parameters of wireless ranging models based on unsupervised learning, which can autonomously learn the characteristics of surrounding environments.

Considering the localization in dynamic wireless signal environments, LEASE [11] deployed dense reference anchors to gather real-time RSSI samples and rebuild the localization model when the observed RSSI features exhibit a significant deviation. In order to reduce the number of reference anchors, Haeberlen et al. [34] adopted the linear function to learn the relationship of RSSIs in proximity, as well as to infer RSSIs for those spots without dedicated reference anchors. J. Yin et al. [35] proposed a model tree-based algorithm to reconstruct the radio map, which takes into account of real-time signal-strength values at each time point and makes use of the dependency between the estimated locations and RPs. In order to counteract the effect of environmental changes and signal noisy, a new feature called soft information (SI) [36] was proposed, which is extracted from intra- and inter-node measurements as well as from contextual data (e.g., digital map, node profile) to realize the accurate localization. Further, F. Morselli et al. [37] presented an SI-based method for fusing heterogeneous observations in 5G ecosystem, in which the DownLink Time-Difference-Of-Arrival (DL-TDOA) measurements from 5G network and time-of-flight (TOF) measurements from WiFi are fused to estimate the location.

Several crowdsourcing based approaches have been designed to update RSSI features in dynamic environments. C. Wu et al. [10] designed a fingerprint updating scheme called AcMu, in which a trajectory matching algorithm is proposed to pinpoint mobile devices' position. Then, a partial least square regression (PLSR) is utilized to learn the relationship between the fingerprint database and RSSI features crowdsourced by mobile devices. S. He et al. [12] proposed a subset-based detection method to identify APs with altered signal and filter them out to maintain the localization accuracy. Then, a Gaussian Process Regression (GPR) is utilized to update the fingerprint with crowdsourced RSSI features. B. Huang et al. [38] established a coarse-grained radio map based on standard GPR, and a marginalized particle extended Gaussian process is leveraged to recursively refine the radio map based on the crowdsourced fingerprints with noisy location labels.

A number of studies have been focused on vehicle indoor tracking. A. Ibisch et al. [39] proposed a vehicle tracking algorithm based on environment-embedded Light Detecting and Ranging (LiDAR) sensors, which leverages Extended Kalman Filter (EKF) to incorporate the locations detected by LiDAR and the vehicle motion model. J. Han et al. [40] proposed a parameterized mapping method for LiDAR to build a computational and storage efficient map that only contains geometric parameters (e.g., diameter and height of a circular column). An EKF is then adopted to fuse the vehicles kinematic model and detected column structures. A. Kumar et al. [41] utilized Convolutional Neural Networks (CNN) to detect the categories and positions of objects in surveillance video of the parking lot. Further, the contour extraction is applied for edge image to enhance the localization accuracy.

A few studies have exploited the inertial sensors embedded in the smartphones for vehicle tracking. R. Gao et al. [26] proposed a smartphone-only system to track vehicles' positions (VeTrack), in which a Sequential Monte Carlo framework is designed to infer the vehicle location based on the map constraints and detected landmarks. C. Jim et al. [25] designed a parking localization system called Parkloc, where a graph-based match algorithm was proposed to project the recorded vehicles' trajectory into the garage map, and a semi-supervised GraphSLAM algorithm was designed to learn the map graph from the observed trajectories. R. Gao et al. [24] proposed the VeTorch, which learns vehicle's moving dependencies from inertial data based on a convolution neural network. A federated learning-based mechanism is further adopted to produce customized models for individual smartphones.

7 Conclusion

This paper proposed an online learning framework for robust vehicle tracking in dynamic indoor parking environments, which consists of an HMM based Online Evaluation method (i.e., HOE) and an Online Transfer Learning algorithm (i.e., OTL). For HOE, the WiFi fingerprinting localization results and the displacement estimated by DR were modelled as the emission probability and the transition probability, respectively. The forward probability was then derived to combine WiFi fingerprint-based and DR-based localization results, and the Wasserstein distance was adopted to measure the discrepancy between the two results. For OTL, two classification models (i.e., the batch and the online models) were combined to estimate the vehicle's locations. Then, a label conversion scheme was designed to build the candidate label set for the online RSSI samples and a weight allocation scheme was proposed to calculate the loss for each model and adjust their weights accordingly. In addition, an instance-based transferring scheme was designed to train the batch model by selecting suitable fingerprints from the offline database. Finally, we implemented the system prototype and gave comprehensive performance evaluation in a realistic environments, which conclusively demonstrated the effectiveness and robustness of the proposed framework.

The current solution relies on the manually obtained map constraints and an initial fingerprinting model, which may still hinder system scalability. In the future work, crowdsourcing-based localization solutions will be further examined to alleviate the initialization overhead. In addition, more efficient fingerprint collection method is expected to be designed for the online learning framework to further enhance the system practicability.

References

[1] T. Lin, H. Rivano, and F. Le Mouël, "A survey of smart parking solutions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3229–3253, 2017.

[2] K. Liu, H. B. Lim, E. Frazzoli, H. Ji, and V. C. Lee, "Improving positioning accuracy using gps pseudorange measurements for cooperative vehicular localization," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 6, pp. 2544–2556, 2013.

[3] H. Jie, K. Liu, H. Zhang, R. Xie, W. Wu, and S. Guo, "Aodc: automatic offline database construction for indoor localization in a hybrid uwb/wi-fi environment," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2020, pp. 324–329.

[4] K. Liu, H. Zhang, J. Ng, Y. Xia, L. Feng, V. Lee, and S. Son, "Toward low-overhead fingerprint-based indoor localization via transfer learning: Design, implementation, and evaluation," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 898–908, 2018.

[5] F. Jin, K. Liu, H. Zhang, J. K.-Y. Ng, S. Guo, V. C. Lee, and S. H. Son, "Toward scalable and robust indoor tracking: Design, implementation, and evaluation," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1192–1204, 2019.

[6] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[7] H. Zhang, K. Liu, F. Jin, L. Feng, V. Lee, and J. Ng, "A scalable indoor localization algorithm based on distance fitting and fingerprint mapping in Wi-Fi environments," *Neural Computing and Applications*, vol. 32, pp. 5131–5145, 2019.

[8] Y. Fu, P. Chen, S. Yang, and J. Tang, "An indoor localization algorithm based on continuous feature scaling and outlier deleting," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1108–1115, 2018.

[9] H. Zhang, K. Liu, Q. Shang, L. Feng, C. Chen, Z. Wu, and S. Guo, "Dual-band wi-fi based indoor localization via stacked denosing autoencoder," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[10] C. Wu, Z. Yang, and C. Xiao, "Automatic radio map adaptation for indoor localization using smartphones," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 517–528, 2017.

[11] P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows, and S. Gamt, "A system for lease: Location estimation assisted by stationary emitters for indoor rf wireless networks," in *IEEE infocom 2004*, vol. 2. IEEE, 2004, pp. 1001–1011.

[12] S. He, W. Lin, and S.-H. G. Chan, "Indoor localization and automatic fingerprint update with altered ap signals," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 1897–1910, 2016.

[13] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 2019 ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL'09)*, 2009, pp. 336–343.

[14] F. Jin, K. Liu, H. Zhang, W. Wu, J. Cao, and X. Zhai, "A zero site-survey overhead indoor tracking system using particle filter," in *Proceedings of the 2019 IEEE International Conference on Communications (ICC'19)*. IEEE, 2019, pp. 1–7.

[15] I. Olkin and F. Pukelsheim, "The distance between two random vectors with given dispersion matrices," *Linear Algebra and its Applications*, vol. 48, pp. 257–263, 1982.

[16] N. Cristianini, J. Shawe-Taylor *et al.*, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[17] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[18] D. Saad, "Online algorithms and stochastic approximations," *Online Learning*, vol. 5, pp. 6–3, 1998.

[19] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[20] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[21] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive aggressive algorithms," 2006.

[22] P. Zhao, S. C. Hoi, J. Wang, and B. Li, "Online transfer learning," *Artificial Intelligence*, vol. 216, pp. 76–102, 2014.

[23] Q. Wu, H. Wu, X. Zhou, M. Tan, Y. Xu, Y. Yan, and T. Hao, "Online transfer learning with multiple homogeneous or heterogeneous sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1494–1507, 2017.

[24] R. Gao, X. Xiao, S. Zhu, W. Xing, C. Li, L. Liu, L. Ma, and H. Chai, "Glow in the dark: Smartphone inertial odometry for vehicle tracking in gps blocked environments," *IEEE Internet of Things Journal*, 2021.

[25] J. Cherian, J. Luo, and S.-S. Ho, "Parkloc: Light-weight graph-based vehicular localization in parking garages," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–23, 2018.

[26] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, and G. Luo, "Smartphone-based real time vehicle tracking in indoor parking structures," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 2023–2036, 2017.

[27] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of the 2000 Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'00)*. IEEE, 2000, pp. 775–784.

[28] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proceedings of the 2015 International Conference on Mobile Systems, Applications, and Services (MobiSys'05)*. ACM, 2005, pp. 205–218.

[29] C.-M. Own, J. Hou, and W. Tao, "Signal fuse learning method with dual bands wifi signal measurements in indoor positioning," *IEEE Access*, vol. 7, pp. 131 805–131 817, 2019.

[30] S. Kumar and S. K. Das, "Target detection and localization methods using compartmental model for internet of things," *IEEE Transactions on Mobile Computing*, vol. 19, no. 9, pp. 2234–2249, 2019.

[31] M. Zhou, Y. Li, M. J. Tahir, X. Geng, Y. Wang, and W. He, "Integrated statistical test of signal distributions and access point contributions for wi-fi indoor localization," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5057–5070, 2021.

[32] M. Z. Win, W. Dai, Y. Shen, G. Chrisikos, and H. V. Poor, "Network operation strategies for efficient localization and navigation," *Proceedings of the IEEE*, vol. 106, no. 7, pp. 1224–1254, 2018.

[33] J. Choi, Y.-S. Choi, and S. Talwar, "Unsupervised learning techniques for trilateration: From theory to android app implementation," *IEEE Access*, vol. 7, pp. 134 525–134 538, 2019.

[34] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, 2004, pp. 70–84.

[35] J. Yin, Q. Yang, and L. M. Ni, "Learning adaptive temporal radio maps for signal-strength-based location estimation," *IEEE transactions on mobile computing*, vol. 7, no. 7, pp. 869–883, 2008.

[36] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, "Soft information for localization-of-things," *Proceedings of the IEEE*, vol. 107, no. 11, pp. 2240–2264, 2019.

[37] F. Morselli, S. Bartoletti, M. Z. Win, and A. Conti, "Localization in 5g ecosystem with wi-fi," in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2021, pp. 441–445.

[38] B. Huang, Z. Xu, B. Jia, and G. Mao, "An online radio map update scheme for wifi fingerprint-based localization," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6909–6918, 2019.

[39] A. Ibisch, S. Stümper, H. Altinger, M. Neuhausen, M. Tschentscher, M. Schlipsing, J. Salinen, and A. Knoll, "Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded lidar sensors," in *2013 IEEE intelligent vehicles symposium (IV)*. IEEE, 2013, pp. 829–834.

[40] J. Han, J. Kim, and D. H. Shim, "Precise localization and mapping in indoor parking structures via parameterized slam," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4415–4426, 2018.

[41] A. K. T. R. Kumar, B. Schäufele, D. Becker, O. Sawade, and I. Radusch, "Indoor localization of vehicles using deep learning," in *2016 IEEE 17th international symposium on a world of wireless, mobile and multimedia networks (WoWMoM)*. IEEE, 2016, pp. 1–6.

Kai Liu (S'07'-M'12-SM'19) received his Ph.D. Degree in Computer Science from the City University of Hong Kong in 2011. From December 2010 to May 2011, he was a Visiting Scholar with the Department of Computer Science, University of Virginia, USA. From 2011 to 2014, he was a Postdoctoral Fellow with Singapore Nanyang Technological University, City University of Hong Kong, and Hong Kong Baptist University. He is currently a Professor with the College of Computer Science, Chongqing University, China. His research interests include Internet of Vehicles, Mobile Computing and Pervasive Computing.

Feiyu Jin received the M.S. Degree in computer science from Chongqing University, Chongqing, China, in 2020, where he is currently pursuing the Ph.D. degree. His research interests include pervasive computing, mobile computing and intelligent transportation system.

Junbo Hu received the B.S. degree in computer science from Chongqing University, Chongqing, China, in 2020, where he is currently pursuing the M.S. degree. His research interests include mobile computing and intelligent transportation system.

Ruitao Xie received her PhD degree in Computer Science from City University of Hong Kong in 2014, and BEng degree from Beijing University of Posts and Telecommunications in 2008. She is currently an assistant professor in College of Computer Science and Software Engineering, Shenzhen University. Her research interests include edge computing, AI networking, cloud computing and distributed systems.

Fuqiang Gu is currently a Professor at the College of Computer Science of Chongqing University, China. He obtained his PhD from the University of Melbourne. Before joining the Chongqing University, he has successively worked as a researcher at the RWTH-Aachen University, University of Toronto, and National University of Singapore. His main research interests are positioning and navigation, robotics, autonomous systems, and machine learning. He is a member of IEEE, ACM, and CCF.

Songtao Guo received his B.S., M.S. and Ph.D. degrees in Computer Software and Theory from Chongqing University, Chongqing, China, in 1999, 2003 and 2008, respectively. He was a professor from 2011 to 2012 at Chongqing University and a professor from 2013 to 2018. At present, he is a full professor at Chongqing University, China. He was a senior research associate at the City University of Hong Kong from 2010 to 2011, and a visiting scholar at Stony Brook University, New York, USA, from May 2011 to May 2012. His research interests include mobile edge computing, mobile cloud computing and wireless ad hoc networks. He has published more than 130 scientific papers in leading refereed journals and conferences. He has received many research grants as a Principal Investigator from the National Science Foundation of China and Chongqing and the Postdoctoral Science Foundation of China.

Jiangtao Luo (M'11-SM'15) received his B.S. and PhD degrees from Nankai University and the Chinese Academy of Science in 1993 and 1998, respectively. Currently, he is a full professor, PhD supervisor and deputy Dean of the Electronic Information and Networking Research Institute at Chongqing University of Posts & Telecommunications (CQUPT). His major research interests are future network architecture, network data mining, visual bigdata analysis and satellite internet. He has published more than 150 papers and owned 30 patents in these fields, actively participated in paper reviewing and conference organizing. He was awarded the Chinese State Award of Scientific and Technological Progress in 2011, the Chongqing Provincial Award of Scientific and Technological Progress twice in 2010 and 2007, respectively, and the Chongqing Science and Technology Award for Youth in 2010. He has been selected as IEEE Senior Member since June 2015 and ACM member since Nov 2013.
His current research interests are future network architecture, visual bigdata analysis and satellite/vehicular Internet.