DeFLoc: Deep Learning Assisted Indoor Vehicle Localization Atop FM Fingerprint Map

Jiale Lei, Junqin Huang, Linghe Kong, *Senior Member, IEEE*, Guihai Chen, and Muhammad Khurram Khan, *Senior Member, IEEE*

***Abstract*—Indoor vehicle localization is an underlying technology for realizing Autonomous Valet Parking (AVP), which demands high accuracy and reliability. However, existing localization technologies, such as GPS, WiFi, Bluetooth, suffer from either low availability or high cost, which are not practical in the real world. In order to put AVP into practice, We desperately need an efficient and reliable indoor vehicle localization technology. In this paper, we propose a Deep learning and FM fingerprint map based indoor vehicle Localization method, namely DeFLoc, which leverages FM signals to achieve accurate and practical indoor localization. In order to reduce the workload of the FM fingerprints collecting process, DeFLoc uses partially uniform sampling to decrease sample data volume and reconstructs the FM fingerprint map from collected incomplete fingerprints precisely using a dedicated deep Convolutional Neural Network (CNN). To alleviate the influence of signal distortions in some FM frequencies, we further design smooth layers in the neural network for improving the accuracy of map reconstruction. Moreover, we devise a continuous vehicle localization algorithm by considering the preferences of vehicle movements to assist us to calibrate localization. We implemented a prototype of DeFLoc and conducted extensive experiments both in simulation and practice. Evaluation results show that our proposed reconstruction model improves accuracy by 40% over conventional matrix completion methods even under the 60% data missing rate. With the precisely reconstructed fingerprint map, DeFLoc achieves over 90% localization accuracy, which indicates DeFLoc can realize accurate and practical indoor vehicle localization.**

***Index Terms*—Autonomous valet parking, convolutional neural network, indoor vehicle localization, fingerprint map reconstruction, FM.**

I. Introduction

AUTONOMOUS Valet Parking (AVP) is emerging with the development of automatic driving. According to different automatic parking capabilities, AVP can be divided into 5 levels [1], where AVP at level 0 refers to totally manual parking. AVP at level 1 supports limited autonomous parking but still requires drivers staying in the vehicle to supervise the parking process. Currently, AVP at level 2 has grown into a proven technique and has been deployed on many commodity intelligent vehicles, which is benefited from the development of various advanced sensors including cameras, Inertial Measurement Units (IMU), and even LiDAR. Vehicles with level 2 AVP automatically seek a nearby parking slot and park themselves into the slot without human interference. AVP at higher levels are expected to realize that drivers just leave their cars at the entrance of the parking area (level 3) or even anywhere in the smart city (level 4), and vehicles will automatically find parking spaces. Although AVP at level 3 is expected to emerge in the near future, a higher level AVP indeed requires more technology breakthroughs to evolve into a proven technique. Given that many parking areas are in indoor environments, a robust indoor vehicle localization technique is critical to bring AVP to a higher level. However, it is non-trivial to achieve accurate indoor vehicle localization, making it one of the bottlenecks that hinder the AVP evolution from level 2 to level 3.

Global Navigation Satellite Systems (GNSS), such as GPS [2], is the most commonly used method for vehicle localization in modern life. However, GPS always performs badly in an indoor environment. Even though there are a variety of Assisted GPS (A-GPS) solutions utilizing cellular networks as an assistance approach for indoor localization [3], [4], they still suffer a large localization error. Since the number of satellites in the line of sight of the vehicle should be no less than three for precise localization, it is difficult to be guaranteed under a partially blocked sky.

One type of indoor localization method is based on short-distance communication, such as WiFi and RFID [5], which have been deeply investigated. However, these methods are still not feasible for improving AVP. Take WiFi as an example, to implement WiFi-based indoor localization, there should deploy adequate WiFi Access Points (AP) that cover the whole parking area. The more APs that vehicles can detect in a preselected position, the higher accuracy the method can achieve. Thus, the deployment of WiFi APs usually has a certain redundancy to guarantee the required localization accuracy. Considering that WiFi is used for communication but not localization, to realize WiFi coverage in large-scale indoor parking areas is a waste of resources. It is also not

Manuscript received August 8, 2021; revised January 5, 2022 and March 4, 2022; accepted March 20, 2022. This work was supported in part by NSFC under Grant 62141220, Grant 61972253, Grant U1908212, Grant 72061127001, Grant 62172276, and Grant 61972254; and in part by the Program for Professor of Special Appointment (Eastern Scholar) at the Shanghai Institutions of Higher Learning. The Associate Editor for this article was B. B. Gupta. *(Corresponding author: Linghe Kong.)*

Jiale Lei, Junqin Huang, Linghe Kong, and Guihai Chen are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: radiumscripttang@sjtu.edu.cn; junqin.huang@sjtu.edu.cn; linghe.kong@sjtu.edu.cn; gchen@cs.sjtu.edu.cn).

Muhammad Khurram Khan is with the Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh 11564, Saudi Arabia (e-mail: mkhurram@ksu.edu.sa).

Digital Object Identifier 10.1109/TITS.2022.3163539

cost-effective to increase the construction cost of parking areas for supporting a higher level AVP.

Another type of indoor localization method utilizes Frequency Modulation (FM) fingerprint to achieve deployment-free indoor localization [6]–[8]. Compared with A-GPS and short-distance-communication-based indoor localization methods, FM-based localization methods cover shortages of low availability and high cost in indoor environments. Firstly, FM-based indoor localization methods utilize public radio stations, which are accessible in an extremely large scope, to construct FM fingerprint maps. These deployment-free methods largely reduce the deployment and maintenance costs of parking area infrastructures. Secondly, the FM receiver is one of the essential devices equipped in various vehicles. Therefore, FM-based indoor vehicle localization requires no extra equipment to be installed on vehicles. Moreover, FM radio signals are more stable and have weaker time-variance than those short-distance communication signals [7]. FM signals usually have strong penetrability, so that they can be detected even in deep underground parking areas.

Although previous works [6]–[8] have made some advances in FM-based indoor localization, they are still not applicable to AVP due to four main deficiencies. Firstly, these works ignored the heavy workload of fingerprint map construction. Given that the fingerprint map requires periodically re-sampling for location calibration, it can be time-consuming to construct and maintain the fingerprint map in the long term. Secondly, some works assumed that FM signals are stable enough so that their features can be described by simple models. However, it is not practical in indoor parking environments because of the existence of multi-path effect and interference. Thirdly, most works were implemented on expensive Software Defined Radio (SDR) devices, such as USRP, which are impossible to appear on vehicles. Moreover, some works sample too few reference points, which are hard to support accurate localization in the large indoor parking area. These limitations make existing FM fingerprint-based localization schemes still not practical in indoor parking scenarios.

In this paper, we propose a **De**ep learning and **F**M fingerprint map based indoor vehicle **Loc**alization method, namely **DeFLoc**, which leverages FM signals to achieve accurate and practical indoor localization. DeFLoc contains two working phases: offline phase and online phase. In the offline phase, an FM fingerprint map would be constructed. In the online phase, FM fingerprints measured by vehicles are matched with the constructed fingerprint map, to determine vehicles' physical positions.

To address the shortcomings in previous works and make DeFLoc practical in AVP, we design several advanced techniques to realize accurate and efficient indoor vehicle localization. Firstly, to reduce the workload of fingerprint map construction in the offline phase, we partially sample fingerprints on each reference point and then reconstruct the fingerprint map from these collected incomplete fingerprints, which can be regarded as a matrix completion problem. However, conventional matrix completion methods suffer large errors and thus cannot meet the high accuracy requirement of AVP. To achieve a precise reconstruction, a dedicated deep Convolutional Neural Network (CNN) is devised to reconstruct the fingerprint map from the collected incomplete fingerprints. Secondly, we design smooth layers in the neural network with considering different kinds of FM frequencies, to further eliminate the effect of FM signal distortions. Thirdly, we also design a continuous vehicle localization algorithm, which takes preferences of vehicle movements in indoor parking areas into account, to reduce the localization error in the online phase. We conclude our main contributions as follows:

- We propose an accurate and practical FM fingerprint-based indoor vehicle localization, named DeFLoc, to promote AVP to a higher level. To our best knowledge, this is the first work that considers the efficiency of FM fingerprint map construction in indoor vehicle localization.
- We design a dedicated deep neural network to reconstruct FM fingerprint map precisely from limited sampling. To alleviate the effect of the multipath and environmental interference, we embed smooth layers in the network according to the features of FM signal profiles. We also design a continuous vehicle localization algorithm to reduce the localization error in the online phase.
- We implemented a prototype of DeFLoc and evaluated the prototype using a dataset collected from the real world. Experiment results show that our proposed method improves reconstruction accuracy by 40% over conventional methods under the 60% data missing rate. Meanwhile, our proposed network model is highly lightweight and can be deployed on edge servers and even terminal devices. What's more, DeFLoc can still achieve more than 90% localization accuracy under the 60% data missing rate.

The rest of the paper is organized as follows. Section II briefly reviews some related works. We give preliminaries and problem definitions in Section III. In Section IV, we introduce the detailed design of DeFLoc. Evaluation settings and results are discussed in Section V. Lastly, we conclude this paper in Section VI.

II. Related Work

We review related works in two aspects: fingerprint-based indoor localization and matrix completion, which are discussed in the following.

A. Fingerprint Based Indoor Localization Technology

Indoor localization technologies mainly can be divided into distance-based and fingerprint-based ones. RADAR [9], Bluepass [10], ArrayTrack [11], and ToneTrack [12] are known as distance-based schemes. Fingerprint-based schemes include Horus [13], HiLoc [14], RadioLoc [8], and so forth.

Experiments have shown that fingerprint-based methods usually achieve more accurate localization [15]. Many fingerprint-based indoor localization methods that use WiFi, Bluetooth or other communication technologies have been proposed in academia.

Currently, WiFi is still the most commonly used technology for indoor localization. For example, Xiong and Jamieson [11] proposed a WiFi-based indoor localization system, named ArrayTrack, which can achieve sub-meter accuracy by utilizing the WiFi Multi-Input Multi-Output (MIMO) property. Wu *et al.* [16] leveraged WiFi Channel State Information (CSI) to describe a WiFi propagation model and build a fingerprint-based localization system. In BarFi [17], barometer sensors were used as the complementary of WiFi to realize more accurate localization. There are some other attempts to improve WiFi localization accuracy through peer assistance [18] or considering the speed of clients [19]. Yang *et al.* [20] also took human impacts on WiFi signals into account and proposed LiFS to eliminate the interference by exploiting user motions from mobile phones.

Besides WiFi, other communication technologies such as RFID, Bluetooth, are also used for indoor localization. For example, Liu *et al.* [21] attached RFID tags on mobile robots to help locate robots themselves. Bluetooth and Bluetooth Low Energy (BLE) are also utilized to locate users by using fingerprint maps [22], [23]. In addition, 4G LTE can provide a rough localization, while it achieves less accuracy compared with the above technologies [24].

Considering the high cost of the deployment and maintenance of infrastructures, *e.g.*, WiFi APs, RFID readers, the aforementioned methods are impractical for large-scale indoor parking areas. Thus, the deployment-free FM-based indoor localization technology has attracted researchers' attention. For example, Chen *et al.* [7] confirmed the feasibility of indoor localization using FM radio through extensive experiments and found that FM radio signal shows weaker time-variance than WiFi. Yoon *et al.* [6] constructed a complicated FM indoor propagation model that considers the changes in indoor environments. Chen *et al.* [8] designed an all-terrain FM-based localization method, called RadioLoc, which considers the impact of multi-path as well as weather conditions. In RadioLoc, a novel data processing framework was designed and implemented on the USRP SDR board, and manually picked features are selected to determine the position.

Although these FM fingerprint-based localization methods are promising, they are still not practical for indoor vehicle localization. Some of these works even rely on the assistance of other signal sources, such as WiFi, to realize an accurate localization, which runs counter to the original intent of deployment-free methods. Besides, they ignore the heavy workload of fingerprint map construction, which can be time-consuming in practice.

B. Matrix Completion

Matrix completion [25] methods have been applied widely for recovering missing data in incomplete matrices. The goal of matrix completion is to minimize recovery error to reconstruct a matrix that is as approximate to the original one as possible.

K-Nearest Neighbors (KNN) [26] is a classical and simple method for matrix completion. It uses the K nearest neighbors of a missing point for interpolation. Singular Value Decomposition (SVD) also works well in matrix completion when the matrix has a low-rank property [27]. SVD decomposes a matrix $M \in \mathbb{R}^{m\times n}$ into a matrix multiplication of three matrices, *i.e.*$M = U\Sigma V^T$, where $U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}$ and $\Sigma \in \mathbb{R}^{m\times n}$. U and V are both orthogonal matrices that satisfy $UU^T = I$ and $VV^T = I$. Σ is a diagonal matrix whose elements are zeros except for those singular values on diagonal. The rank of a matrix is exactly the number of its non-zero singular values. When the rank of M is low, the corresponding Σ has very few non-zero values on its diagonal. The SVD-based method is common in matrix completion. For example, Iterative SVD [28] method was proposed to complete the matrix with noise; Mazumder *et al.* [29] extended SVD by leveraging iterative soft threshold.

With the rapid development of deep learning, researchers begin to explore Deep Neural Networks (DNN) for matrix completion [30], [30], [31]. Deep learning based methods have shown strength in matrix completion. Since a deep network structure has a great characterization ability and supports non-linear transformations in a tensor level, which covers the shortages of conventional matrix completion methods.

Matrix completion methods are used in many real scenarios, including video stream recovery [32], [33], compressive sensing [34], recommender system [35], and transfer learning [36]. Researchers also apply matrix completion on fingerprint map reconstruction. Liu *et al.* [37] proposed a two-phase adaptive sampling strategy by analyzing at a tensor level. Gu *et al.* [38] combined Sparsity Rank Singular Value Decomposition (SRSVD) with KNN algorithm to recover WiFi fingerprint map and thus reducing fingerprint collection. Cheng *et al.* [39] designed a mobile indoor localization scheme that adopts a matrix completion approach to efficiently utilize collected information.

These approaches are mainly designed for WiFi-based indoor localization. Due to the stronger multi-path effect of FM compared to WiFi, the reconstruction of FM fingerprint maps requires a dedicated design. Many works on fingerprint map reconstruction are based on low-rank assumptions of fingerprint maps. However, FM fingerprint map may not show a perfect low-rank property. Fig. 1 shows the ordered singular values of an FM fingerprint map collected in our experiment. Note that the y-axis is on a logarithmic scale. The first several singular values are extremely large, and the rest of the singular values are non-zero, which is known as the long-tail effect. This reveals the reason that conventional matrix completion methods do not perform ideally on recovering FM fingerprint maps.

III. Background and Definition

In this section, we first introduce the background knowledge of FM-fingerprint-based indoor localization, and then we formulate the localization problem mathematically.

A. FM Fingerprint Based Indoor Localization

Different from distance-based localization, the outputs of fingerprint-based localization are usually discrete points rather

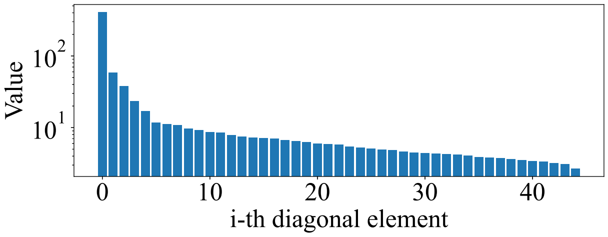

Fig. 1. Ordered singular values of an FM fingerprint map.

than precise coordinates. These points are called reference points, which are defined as follows:

Definition 1: A **reference point** p^j is a selected physical reference position, where the fingerprints are to be collected to form the fingerprint map. $\mathcal{P} = \{p^j \mid j = 1, 2, \ldots, N_P\}$ denotes all reference points, where N_P is the total number of selected reference points.

Fingerprint is another important concept, which describes the FM radio signal feature on a reference point p^j. In DeFLoc, we use Received Signal Strength (RSS) to represent the fingerprint. RSS is a basic measurement in all kinds of signals such as WiFi, Bluetooth, and so forth. It is a logarithmic function of received signal power and calculated by the following equation:

$$RSS = 10 \cdot log(P), \tag{1}$$

where P denotes the power of received signals on a point. Note that RSS value is usually a negative value, and some devices provide Received Signal Strength Indicator (RSSI) value that is a transformation of RSS value.

FM radio usually covers frequencies from 78 MHz to 110 MHz, and public FM radio stations scatter in this frequency range. We divide this range into N_V frequency points υ_j, which is represented as $\mathcal{V} = \{\upsilon_j \mid \upsilon_j = 1, 2, \ldots, N_V\}$. By scanning these frequencies, one can obtain an RSS value at each FM frequency. The vector of RSS values forms a fingerprint at a certain position. We give the definitions of fingerprint and fingerprint map as follows:

Definition 2: A **fingerprint** $f(p)$ at position p is a column vector of RSS values measured at preset frequencies, *i.e.*, $f(p) = (r^{\upsilon_1}(p), r^{\upsilon_2}(p), \ldots, r^{\upsilon_{N_V}}(p))^T$, where $r^{\upsilon_j}(p)$ refers to measured RSS value at frequency υ_j.

Definition 3: A **fingerprint map** is a matrix $\mathcal{X} \in \mathbb{R}^{N_P \times N_V}$ that combines fingerprints at all reference points and is represented as $\mathcal{X} = (f(p^1), f(p^2), \ldots, f(p^{N_P}))^T$.

The first sub-figure in Fig. 2 shows a piece of FM fingerprint collected by using a USRP N210 as the receiver. We totally collected hundreds of samples, and calculated their variances at each frequency, as shown in the second sub-figure in Fig. 2. We observe that there are two RSS peaks in Fig. 2, which indicate two FM radio channels at 89.8MHz and 91.3 MHz, respectively. We define the station frequency as follows:

Definition 4: A **station frequency** υ^S is a FM radio frequency where a radio station transmits. All station frequencies are denoted as $\mathcal{V}^S = \{\upsilon^S_j \mid j = 1, 2, \ldots, N^S_V\}$. Note that we have $\mathcal{V}^S \subset \mathcal{V}$.

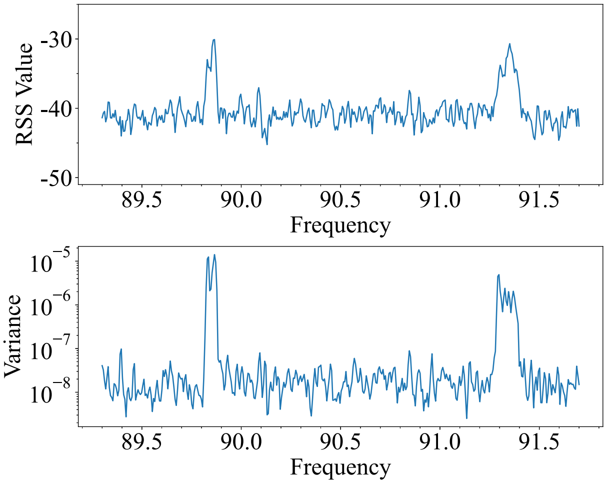

Fig. 2. A piece of collected FM fingerprint ranging from 81.3 MHz to 91.7 MHz, which contains two FM radio station frequencies.

Due to the multi-path effect, we can also observe a large variance peak appearing beside the station frequency. Frequencies between two station frequencies carry stable noise with low variance. We call these frequencies noise frequencies, which are defined as follows:

Definition 5: A noise frequency υ^N is far from station frequency and carries stable background noise. All noise frequencies are denoted as $\mathcal{V}^N = \{\upsilon^N_j \mid j = 1, 2, \ldots, N^N_V\}$. Similarly, we have $\mathcal{V}^{\mathcal{N}} \subset \mathcal{V}$.

In general, a fingerprint-based indoor localization scheme contains two phases: offline phase and online phase. In the offline phase, fingerprints at all reference points $\mathcal{P}$ are collected to form the fingerprint map $\mathcal{X}$. In the online phase, a vehicle measures current position's fingerprint $\hat{f}$, matches its fingerprint with the fingerprint map $\mathcal{X}$, and outputs an inferred vehicle position $\hat{p}$ according to the matching result. We can see that these two phases play different roles in the localization process. The offline phase provides the fingerprint map that serves as a navigator for the online phase. Since the collection of fingerprint map is workload-heavy and those fingerprints do not change very frequently, the fingerprint map only needs to update periodically. Thus, the collection of the fingerprint map belongs to the offline phase. Given the collected fingerprint map, vehicles can easily localize their positions in real-time by comparing measured fingerprints, which belongs to the online phase.

B. Problem Definition

Now we formulate the indoor localization problems to be solved and claim the goals to achieve. Since a configuration process is needed in advance, the localization can be divided into offline phase and online phase.

Offline Phase: We first select a set of reference points $\mathcal{P}$ that contains N_P reference points in an indoor parking area. Let $\mathcal{X}$ denote the ground truth fingerprint map of $\mathcal{P}$. To reduce the fingerprint sampling workload, we utilize the sampling function S to sample an incomplete fingerprint map $\mathcal{Y}$, *i.e.*, $\mathcal{Y} = S(\mathcal{X}, \alpha)$, where $\alpha \in (0, 1)$ denotes the missing

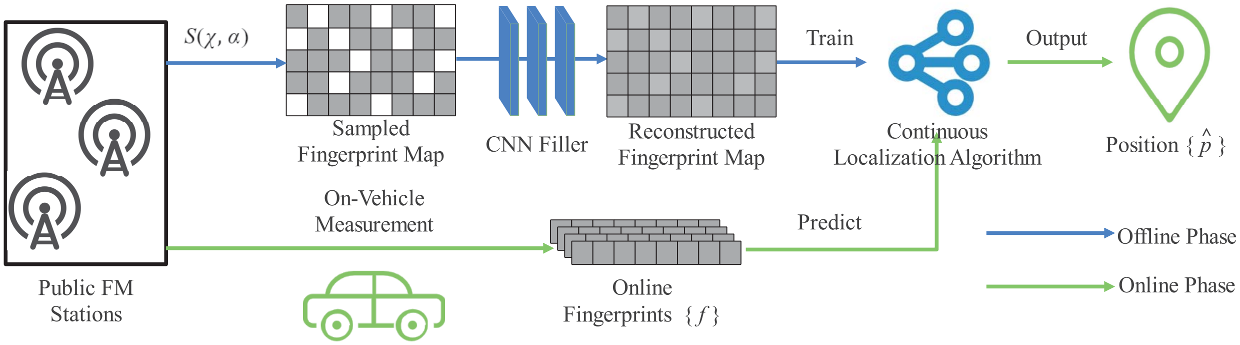

Fig. 3. The framework design and workflow of DeFLoc.

rate of sampling. It is evident that a larger α reduces more sampling workload but makes it more difficult to reconstruct the fingerprint map in the meantime. The sampling function S is actually a linear function of $\mathcal{X}$ and can be defined as follows:

$$S(\mathcal{X}) = \mathcal{X} \odot \mathcal{M}^{\alpha}_{mask}, \tag{2}$$

where $\odot$ denotes the Hadamard product (known as element-wise production), $\mathcal{M}^{\alpha}_{mask}$ is the preset sampling mask matrix that contains only 0 and 1. To reconstruct the fingerprint map, we define the recovery function R to obtain a recovered fingerprint map $\mathcal{X}'$ by operating $\mathcal{X}' = R(\mathcal{Y})$.

Online Phase: The online localization is based on the reconstructed fingerprint map $\mathcal{X}'$. For a fingerprint $f(p)$ collected by the vehicle at position p, we have a localization function L that matches $f(p)$ with the reconstructed fingerprint map $\mathcal{X}'$, and determines the predicted position $\hat{p}$, *i.e.*, $\hat{p} = L(\mathcal{X}', f(p))$.

We can observe that the localization accuracy relies heavily on the quality of the reconstructed fingerprint map. Therefore, it is critical to ensure the reconstruction performance of R. That is to say, The goal of the map reconstruction should be minimizing the reconstruction error, *i.e.*,

$$\min_{R} ||R(\mathcal{Y}) - \mathcal{X}||_F^2, \tag{3}$$

where $||\cdot||_F$ denotes the Frobenius norm of a matrix. Moreover, to achieve accurate localization, it is critical to minimize the localization error, *i.e.*,

$$\min_{L} D(p, L(\mathcal{X}', f(p))), \tag{4}$$

where $D(\cdot,\cdot)$ denotes the physical distance between two positions.

IV. Deep Learning and FM Fingerprint Map Based Indoor Vehicle Localization

The workflow of DeFLoc is illustrated in Fig. 3. In the offline phase, we periodically sample FM fingerprints with a certain missing rate in indoor parking areas. Then the collected incomplete fingerprint map is recovered by our proposed dedicated CNN model to get the reconstructed fingerprint map. In the online phase, a sequence of on-vehicle collected fingerprints is fed to our continuous localization algorithm to output the predicted positions. In the following sections, we will describe the designs of the uniform sampling method, the reconstruction CNN model, and the continuous localization algorithm in detail, respectively.

A. Uniform Sampling

Although fingerprint map collection is usually done by robots, collecting the whole fingerprint map is still time-consuming. To reduce the workload of fingerprint collection in the offline phase, we avoid collecting the whole fingerprint map $\mathcal{X}$ on $\mathcal{P}$. Instead, we partially sample from $\mathcal{X}$, and then the fingerprint map is reconstructed according to the propagation model.

However, FM propagation model is hard to build precisely in an indoor parking environment, since it contains both outdoor and indoor propagation signals and suffers the multipath effect. Moreover, previous works are usually implemented on high-end receiver devices, such as USRP. These devices have very large bandwidths that can cover the whole FM frequency range, which does not work for on-vehicle low-end devices. Most low-end FM devices have very small bandwidths so that RSS at each frequency can be only measured one by one.

Therefore, we sample the FM fingerprints using partial uniform sampling in a frequency-wise way. For each position p, we randomly mark a subset $\mathcal{V}^{D_p} \subset \mathcal{V}$ as missing frequencies according to the preset missing rate α, *i.e.*, $\frac{|\mathcal{V}^{D_p}|}{|\mathcal{V}|} = \alpha$. Then we only measure RSS values at frequencies set $\mathcal{V} \setminus \mathcal{V}^{D_p}$ and fill the rest RSS values with zero.

It is worth mentioning that $\mathcal{V}^{D_p}$ is related to position p. That means FM fingerprints at different positions have different $\mathcal{V}^{D_p}$, as shown in Fig. 3, where white blocks represent missing RSS values. Otherwise, if all fingerprints share the same $\mathcal{V}^{D_p}$, reconstructing the fingerprint map can be less feasible.

B. Fingerprint Reconstruction CNN

To reconstruct the fingerprint map, we leverage a convolutional neural network as the map filler. For FM signal, the distribution of RSS values at station frequency and noise frequency can be distinct. Therefore, we design smooth layers that train parameters respectively for station frequency and

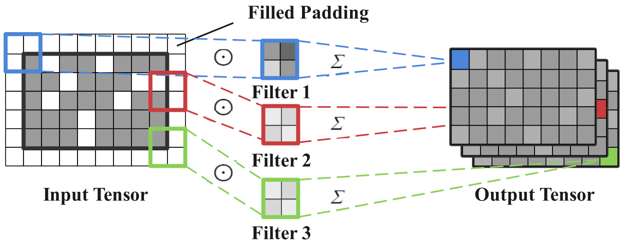

Fig. 4. An example of 3-filter Conv2D layer.

noise frequency, to better model the features of the fingerprint map.

1) Convolutional Layer: To reconstruct the fingerprint map from an incomplete one, we design a dedicated CNN as the filler. Deep neural network has a great ability to fit a function through learning. Given that the fingerprint map is stored in a matrix form, we adopt 2-Dimensional Convolutional (Conv2D) layer in the proposed deep neural network to fulfill the task of fingerprint map reconstruction.

A Conv2D layer contains several filters. It is actually a linear function that takes the input tensor and maps it to the output tensor by calculating the convolution between a filter and a subsection of the input tensor. Note that here tensors are 3-dimensional and the shape of tensors can be represented by a tuple $(width, height, depth)$.

Fig. 4 shows an example of how an output tensor is obtained in a Conv2D layer. The input tensor is firstly filled with empty padding according to the size of filters so that the output tensor shares the same width and height as the input tensor. After filling the padding, every combination of a filter-shape subsection T_i^I in the input tensor and a filter F_j generates a corresponding target value $T_{i,j}^O$ in the output tensor. The target value is computed as follows:

$$T_{i,j}^O = \sum (T_i^I \odot F_j), \tag{5}$$

where $\sum(\cdot)$ means the sum of all elements in the tensor. Note that the Hadamard product is calculated between two tensors that share the same shape, which means the input tensor and the filters share the same depth. As illustrated in Fig. 4, the depths of both input tensor and filters are 1, and the depth of output tensor is equal to the number of filters. It is worth mentioning that the size of the filter should be relatively small in our network since two reference points that are farther apart have less correlation.

With Conv2D layers, we construct a deep convolutional neural network as shown in Fig. 5. The proposed network has a symmetric structure that contains only convolutional layers and smooth layers (smooth layer is introduced in Section IV-B.2). As mentioned before, the thickness of the Conv2D layer represents the number of filters it contains. In our design, we set the layer *Conv1, Conv2, Conv3* in Fig. 5 with $16, 32, 64$ filters, respectively. To introduce non-linearity into deep neural network, we adopt Rectied Linear Units (ReLU) as the activation function, which is defined as $g(x) = max(0, x)$.

A single Conv2D layer works like KNN since both of them consider its several neighbors. But it makes sense when the network grows deep. A deep convolutional neural network has a strong learning ability and can find a natural representation of the fingerprint map via training. Therefore, it is beneficial for improving reconstruction precision.

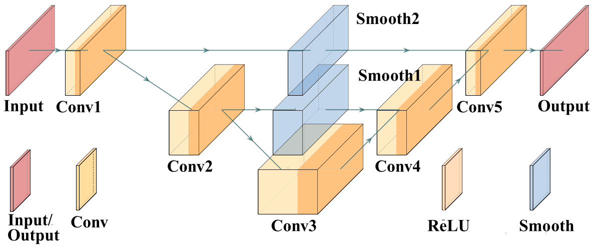

Fig. 5. Structure of proposed CNN model for fingerprint map reconstruction.

2) Smooth Layer: To accelerate training of the network and improve the network performance in the meantime, we add *smooth layers* in the network (shown as the blue layers in Fig. 5), which utilize the features of FM fingerprints. As defined in Def. 4 and Def. 5, RSS values at noise frequencies usually have less variance and hence are more stable, while RSS values at station frequencies have larger mean and variance. Therefore, we design the smooth layers that consider both kinds of frequencies and smooth their RSS values with trainable weights.

We set weights w^S and w^N for station frequency and noise frequency, respectively. Each weight has the same shape of input tensor and only works on its corresponding frequencies. The smooth layer can be regarded as the following function:

$$s(T^I) = w^S \odot T^I[:, \mathcal{V}^{\mathcal{S}}, :] + w^N \odot T^I[:, \mathcal{V}^{\mathcal{N}}, :], \tag{6}$$

where $T^I[:, \mathcal{V}^{\mathcal{S}}, :]$ and $T^I[:, \mathcal{V}^{\mathcal{N}}, :]$ are the lateral slices of the input tensor that are corresponding to $\mathcal{V}^{\mathcal{S}}$ and $\mathcal{V}^{\mathcal{N}}$, respectively. These two parameters will be updated during the training process of the network.

The smooth layer contains trainable variables directed on both station frequencies $\mathcal{V}^{\mathcal{S}}$ and noise frequencies $\mathcal{V}^{\mathcal{N}}$. RSS values of these frequencies have less variance over different positions so that they can be further modeled by smooth layers. We remain RSS values on frequencies except these two types reconstructed by the Conv2D layer since Conv2D layer has a stronger representation ability.

3) Network Training: The deep neural network is trained using the Back-Propagation (BP) algorithm. The error loss function is employed to measure the difference between the ground truth and the output of the network, which is defined as

$$\begin{aligned} J(\theta) &= ||\hat{\mathcal{X}} - \mathcal{X}||_F^2 \\ &= \sum_i \sum_j (\hat{\mathcal{X}}[i,j] - \mathcal{X}[i,j])^2, \end{aligned} \tag{7}$$

where $\hat{\mathcal{X}}$ is the recovered fingerprint map output by the network, and $\mathcal{X}$ is the ground truth fingerprint map. θ represents trainable parameters in the CNN model. By minimizing the value of loss function $J(\theta)$ with BP algorithm, the weights in Conv2D layers and smooth layers are updated with Adaptive Moment Estimation (Adam) optimizer until the value of $J(\theta)$ converges.

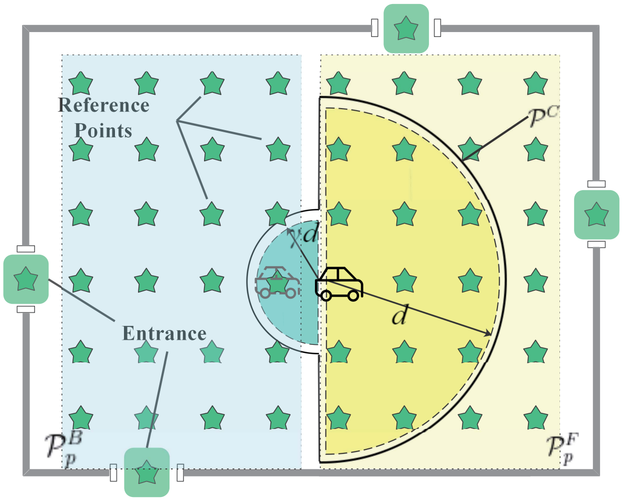

Fig. 6. Composition of candidate positions $\mathcal{P}^C$.

C. Continuous Localization Algorithm

To achieve accurate localization, we design the continuous localization algorithm to determine the position of vehicles. We begin with analyzing the movement of vehicles in an indoor parking area during its AVP process, *i.e.*, a vehicle firstly enters the parking area from one entrance, and drives itself to the allocated parking slot following the roads at a low speed. The movement status of a vehicle at one moment usually can be categorized as three types: ***Moving forward*** (including turns), ***Staying*** or ***Moving backward***. And the third status almost only appears when the vehicle is driving into a parking slot, which is already realized in AVP level 2. Therefore, we only consider the first two movement status.

To avoid matching an online fingerprint with the whole fingerprint map, we set a candidate position set $\mathcal{P}^C$ and corresponding fingerprint set $\mathcal{F}^C$ for the vehicle to determine the possible positions. $\mathcal{P}^C$ is initially set as $\mathcal{P}^E$, which denotes the collection of reference points located near entrances. The online fingerprint is then matched with $\mathcal{F}^C$ rather than the whole fingerprint map. We simply adopt cosine similarity for fingerprint matching, which is calculated as follows:

$$cosine(x, y) = \frac{\langle x, y\rangle}{||x|| \cdot ||y||}, \tag{8}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors, and $||\cdot||$ denotes the L2 norm of a vector. It is evident that the two most similar fingerprints have the largest cosine similarity.

Suppose the first localization output is p, we then update $\mathcal{P}^C$ and $\mathcal{F}^C$ according to p. We use $\delta(p, \mathcal{P}^V)$ as the new $\mathcal{P}^C$, where $\mathcal{P}^V$ denotes the set of history localization outputs. Fig. 6 shows the composition of updated $\mathcal{P}^C$. $\delta(p, \mathcal{P}^V)$ considers those reference points close to p and the potential moving directions of the vehicle coming from the position $\mathcal{P}^V$. Reference points are usually selected in grid, therefore, p has at most 4 directions. Given that vehicles seldom move backward during the process of looking for an available parking slot, reference points on the opposite moving direction of the vehicle will be unlikely the next point where the vehicle will appear. Suppose $\mathcal{P}_p^B$ denotes a set of reference points that are on the opposite direction of $\mathcal{P}^V$ from p (backward direction), and $\mathcal{P}_p^F = \mathcal{P} \backslash \mathcal{P}_p^B$, (forward direction). We formulate $\delta(p, \mathcal{P}^V)$ as

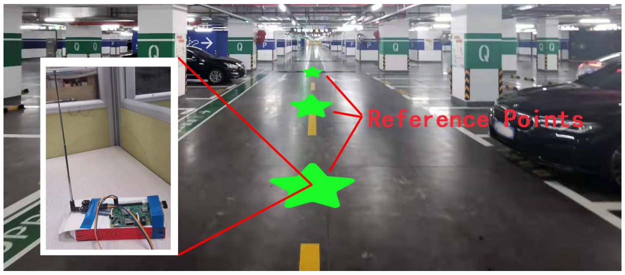

Fig. 7. Environment setup: data collection devices and an indoor parking area. Green stars represent reference points.

$$\begin{aligned}\delta(p, \mathcal{P}^V) = \{&p' \mid D(p, p') < \gamma d, p' \in \mathcal{P}_p^B\} \\ &\cup \{p' \mid D(p, p') < d, p' \in \mathcal{P}_p^F\},\end{aligned} \tag{9}$$

where $D(\cdot, \cdot)$ denotes the distance between two reference points, d is the neighbor radius and $\gamma \in (0, 1)$ is the opposite direction elimination rate. A larger γ can tolerate a larger error of the output p. A larger d is also more error tolerant, since the scale of candidate positions grows larger. However, too large d can make it more complex for fingerprint matching, and thus increases the probability of wrong matches.

In algorithm 1 we show the complete algorithm. Note that each iteration only requires current f^v and previous information, so that the result of the current iteration can be output immediately (seen as **yield**).

V. Evaluation and Analysis

A. Experiment Setting

1) Data Collection Device: In our experiment, we used a TEA 5767 FM module as the FM receiving device. TEA 5767 FM module is a commonly used FM receiver and similar to on-vehicle FM devices that have limited bandwidth and low sensitivity. A TEA 5767 board can support frequencies ranging from 76 MHz to 108 MHz. It provides a 4-bit RSSI value (*i.e.*, 0-15) to measure the received signal strength.

The setting of data collection is shown in Fig. 7. The FM module is connected to a Raspberry Pi board using I^{2}C communication. The Raspberry Pi is programmed to control the FM module, including setting the FM module to a new frequency and reading the RSSI value periodically. The Raspberry Pi is controlled by a PC using SSH protocol. Collected data are temporarily stored in the Raspberry Pi and further uploaded to the PC. To get more accurate RSSI values, we collect RSSI values at a certain frequency 4 times and calculate their average values as the final recorded RSSI values.

2) Field Setup and Data Collection: We chose an indoor underground parking area as our experiment field. The parking area contains three floors, numbered B1, B2, and B3. We chose B3, the deepest floor, to collect data. Fig. 7 shows the B3 floor where we collected data.

The plane structure of our chosen parking area is shown in Fig. 8. The parking area contains many parking blocks

Algorithm 1 Continuous Localization $L(\mathcal{F}^V)$

```
Input: A sequence of fingerprints collected by vehicle $\mathcal{F}^V$
Output: A sequence of predicted position $\mathcal{P}^V$
// Initiate $\mathcal{P}^C$, $\mathcal{F}^C$ from entrance
   reference points.
$\mathcal{P}^C \leftarrow \mathcal{P}^E$
$\mathcal{F}^C \leftarrow \{f(p) \mid p \in \mathcal{P}^C\}$
$\mathcal{P}^V \leftarrow \{\}$
for $f^v$ in $\mathcal{F}^V$ do
    // Match $f^v$ with $\mathcal{F}^V$ using cosine
       similarity.
    $t \leftarrow \{\}$
    for $(p^c, f^c)$ in $(\mathcal{P}^C, \mathcal{F}^C)$ do
        $s \leftarrow cosine(f^v, f^c)$
        $t \leftarrow t \cup \{(p^c, s)\}$
    end
    // Find and output matched reference
       point.
    $(p, s) \leftarrow \max_s t$
    yield $p$
    $\mathcal{P}^V \leftarrow \mathcal{P}^V \cup \{p\}$
    // Update $\mathcal{P}^C$, $\mathcal{F}^C$ from current
       predicted position $p$ and history
       track $\mathcal{P}^V$.
    $\mathcal{P}^C \leftarrow \delta(p, \mathcal{P}^V)$
    $\mathcal{F}^C \leftarrow \{f(p) \mid p \in \mathcal{P}^C\}$
end
return $\mathcal{P}^V$
```

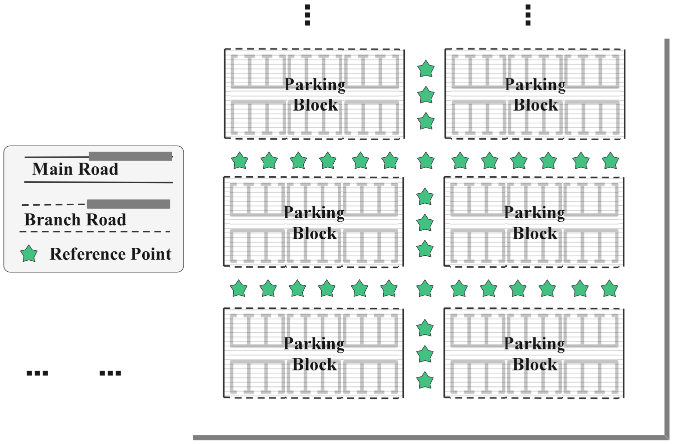

Fig. 8. Plane structure of the indoor parking area.

which are separated by main roads and branch roads. Since our localization is to help vehicles navigate themselves in the parking area, we select all reference points on roads instead of parking blocks, as green stars illustrated in Fig. 8. According to the recommended setting in RadioLoc [8], we set the distance of two adjacent reference points as 4.8 meters, which is close to the length of a car and exactly equal to the width of two parking slots.

In our experiment field, the frequencies of FM radio stations scatter from 87 MHz to 107 MHz. Therefore, we chose this range to collect FM fingerprints. We set 100 kHz as the interval length and divided the range 87 MHz to 107 MHz into 201 frequencies, *i.e.*, $\{87.0, 87.1, \ldots 106.9, 107.0\}$ MHz. Therefore, the length of an FM fingerprint is 201. We totally chose 4 subareas in the parking area. In each subarea, we selected 45 reference points to collect fingerprints. Each subarea covers about 3000 m^2 and contains more than 150 parking slots. Then a fingerprint map in our experiment is a 45×201 matrix. Note that we collected complete fingerprint maps because we need the ground truth to evaluate the performance of our system. Information about collected fingerprints is shown in Tab. I. We further conducted 5-fold cross-validation [40] where we divided the dataset into the training set, validation set, and test set with the proportion of 80%, 10%, and 10% respectively.

TABLE I

Information About Collected Fingerprints

# subarea	number of fingerprints
1	450
2	270
3	270
4	270
Total	1260

TABLE II

Parameters of Implemented Neural Network

# layer	output shape	filters	filter size
Input	(45, 201, 1)	16	(2, 1)
Conv1	(45, 201, 16)	16	(2, 1)
Conv2	(45, 201, 32)	16	(2, 2)
Conv3	(45, 201, 64)	16	(1, 4)
Conv4	(45, 201, 32)	16	(2, 2)
Conv5	(45, 201, 16)	16	(2, 1)
Smooth1	(45, 201, 36)	-	-
Smooth2	(45, 201, 16)	-	-
Output	(45, 201, 1)	-	-

3) System Implementation: We built our system on a PC with AMD Ryzen 3600X CPU, 16G RAM, and Nvidia RTX 2060 GPU. We implemented the proposed CNN network using Keras, which is a deep learning library taking Tensorflow as the backend. According to our collected data, we determined the parameters of our neural network, as shown in Tab. II.

We totally set 6 data missing rates, from 10% to 60%, to evaluate our fingerprint reconstruction algorithm extensively. Since we collected complete fingerprints in the data collection phase, we uniformly removed a part of values in collected fingerprints as the inputs of the CNN network. And we used the corresponding complete ones as the ground truth to train the network. Note that we trained a network for each missing rate using mixed data from 4 subareas. By monitoring the loss change on the validation set, the early stop mechanism stops the training in time to avoid model overfitting. The learning rate was set as 3×10^{-3} with a decay of 10^{-5}.

For testing the online localization accuracy, we simulate totally 320 vehicle tracks in 4 subareas based on the ground truth fingerprints. Each track starts from one of the entrances. For each movement, the vehicle moves forward with an 85% possibility or stays with a 15% possibility. To simulate the interference in the online phase, we also add Gaussian noise to our generated fingerprint sequences of each track.

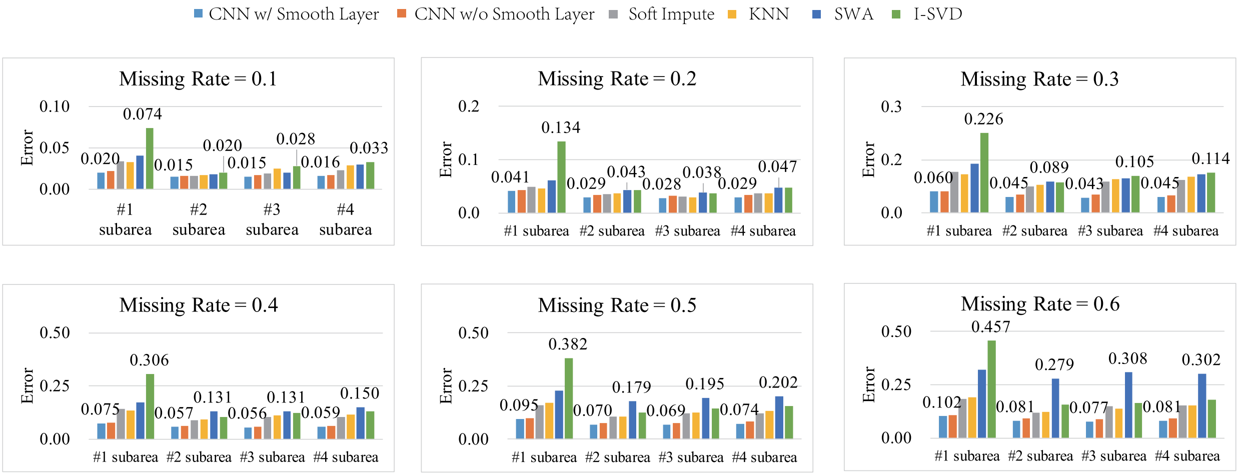

Fig. 9. Reconstruction error of several methods under different data missing rates.

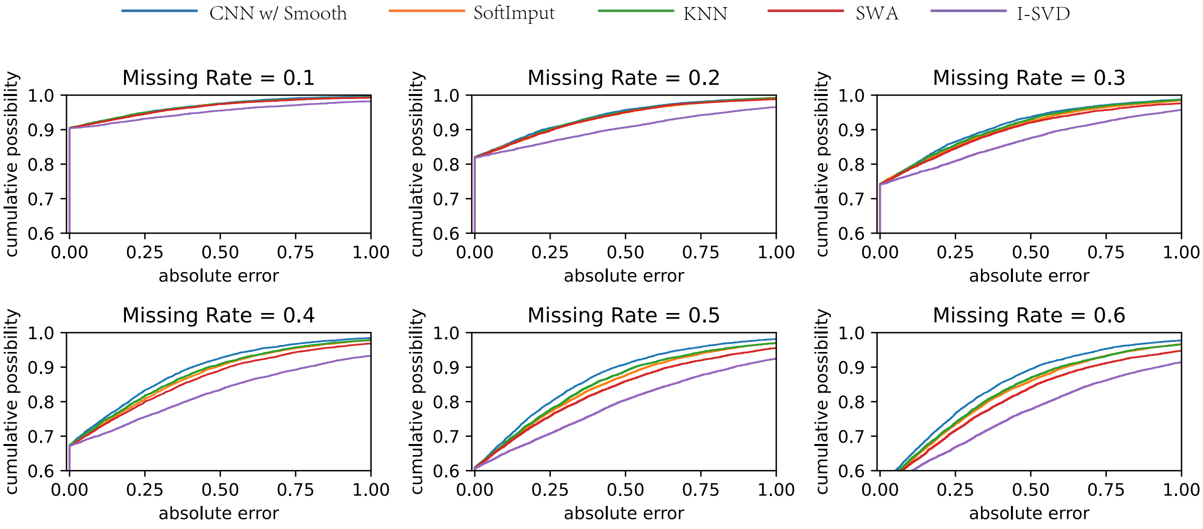

Fig. 10. CDF of mean absolute error of each method under different data missing rates.

The simulated noise is 1.5x larger than the noise in the collected dataset. Our continuous localization algorithm was then evaluated on these simulated tracks using the reconstructed fingerprint map obtained in the previous stage.

4) Metrics: We measure the performance of fingerprint map reconstruction by Mean Squared Error (MSE), *i.e.*,

$$\frac{1}{N_P \times N_V} \sum_{i=1}^{N_P} \sum_{j=1}^{N_V} (\mathcal{X}[i, j] - \hat{\mathcal{X}}[i, j])^2, \tag{10}$$

where $\mathcal{X}$ is the ground truth fingerprint map and $\hat{\mathcal{X}}$ is the output of reconstruction methods. In this experiment, we have $N_P = 45$ and $N_V = 201$.

We use the accuracy and error rate to measure the performance of localization methods. For the i-th track, we define the localization accuracy as

$$acc_i = \frac{TP_i}{TP_i + FP_i}, \tag{11}$$

where TP_i and FP_i are the numbers of true positive and false-positive matching, respectively. Since there are multiple simulated tracks in the experiment, we calculate the average accuracy as the final accuracy of the methods.

Suppose the length of the i-th track is l_i, and the j-th output in the i-th track is denoted as $\hat{p}_{i,j}$. Its ground truth is denoted as $p_{i,j}$. The localization error of the i-th track is defined as

$$e_i = \frac{1}{l_i} \sum_{i=1}^{l_i} D(p_{i,j}, \hat{p}_{i,j}). \tag{12}$$

Similarly, we take the mean of error values of all tracks as the final error value, called Average Localization Error (ALE).

B. Performance of Fingerprint Reconstruction CNN

1) Reconstruction Precision: We evaluated the performance of our fingerprint reconstruction network under 6 missing rates and 4 subareas. Since we firstly propose to reconstruct FM fingerprint map using deep learning method, few methods can be directly applied in this scenario. Therefore, we apply some classical general matrix completion methods including KNN, Iterative SVD (I-SVD), SoftImpute, and Similar Weighted Averaging (SWA) supported by the open-source library [41] on our collected dataset. To validate the effectiveness of our proposed smooth layer, we also trained networks without smooth layers.

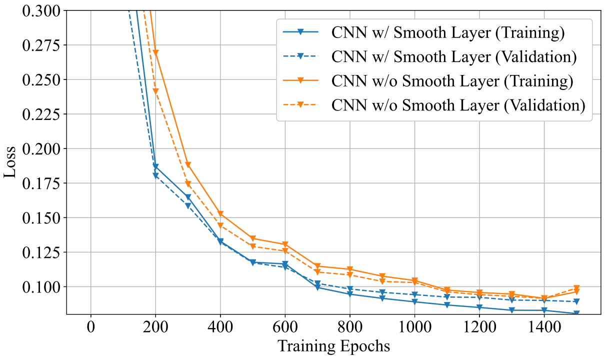

Fig. 11. Training process when data missing rate = 0.5.

Fig. 9 shows the performance of each reconstruction method measured by MSE. When the missing rate is less than 20%, MSE of each method rises with the increasing data missing rate. For instance, the error of our proposed CNN with smooth layers is no more than 0.021, when the missing rate is 10%. And the MSE increases to 0.109 when the missing rate comes to 60%. Besides, when the missing rate is less than 20%, all methods have acceptable reconstruction accuracy. But when the missing rate is larger than 20%, our proposed model has an obvious advantage over other conventional methods. Even under a missing rate of 60%, our reconstruction error is over 40% less than SoftImpute which achieves the best performance among these conventional methods. In addition, in most cases, CNN with smooth layers outperform that without smooth layers slightly, which shows that smooth layer can further improve the reconstruction performance.

We also calculate the Cumulative Distribution Function (CDF) of mean absolute error of each reconstruction method, which is illustrated in Fig. 10. When the missing rate is less than 30%, all methods show good CDF curves except I-SVD. And when the missing rate is larger than 30%, the CDF curve of our proposed network is always over other curves, which means our proposed network outperforms other methods. The advantage is more evident under a larger missing rate. Even with a missing rate of 50% or 60%, our proposed CNN with smooth layer can keep the mean absolute error smaller than 0.5 with a 90% possibility.

These results show that our proposed fingerprint reconstruction model has a strong learning ability and can be qualified for FM fingerprint map reconstruction.

2) Performance of Smooth Layer: In the experiment, we trained our proposed CNN model on an Nvidia RTX 2060 GPU. We set the maximal training epochs as 3000 for each network. With the early stop mechanism, it takes much fewer epochs than maximal epochs to finish training.

Fig. 11 shows the training process when the missing rate is 50%. The figure shows that the loss of both networks (w/ or w/o smooth layers) converges with training epochs increasing. Nevertheless, the network with smooth layers shows better performance than that without smooth layers both on training set and validation set. Moreover, during the training process, we take the change of loss on validation set as the indicator of convergence. Therefore, we observe that network with smooth layers converges during 900 to 1000 epochs, while network without smooth layers converges during 1100 to 1200 epochs. Since smooth layers learn the features of those low-variance frequencies specifically, they can make the training process converges faster. In conclusion, our proposed CNN model with smooth layers addresses the challenges of low precision of fingerprint map reconstruction and signal distortions.

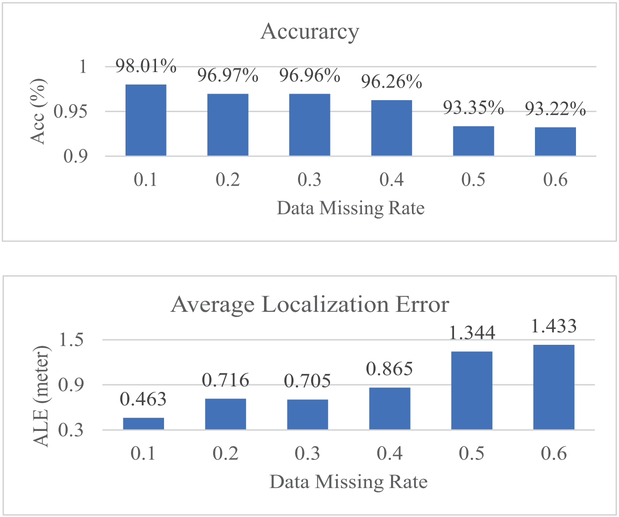

Fig. 12. Localization accuracy and ALE under different data missing rates.

Our proposed network is also efficient both in training and predicting. It takes less than 40 milliseconds to train the network for one epoch. Therefore, the training process can be finished within minutes. And in the predicting phase, one prediction can be finished within 100 milliseconds. In addition, the size of our model is less than 6 MB, which is lightweight enough to be deployed on edge servers or even terminal devices.

C. Performance of Continuous Localization Algorithm

In our simulation, we mainly tested the accuracy and ALE of our proposed continuous localization algorithm. By using a dynamic local candidate fingerprint set instead of a global one, our algorithm reduces the workload for matching and meanwhile achieves high accuracy. The accuracy and ALE under different missing rates are shown in Fig. 12. The missing rate does affect both the accuracy and ALE of the localization. Especially when the missing rate is larger than 40%, the performance of localization decreases sharply. This is reasonable since precise localization relies on the high quality of the fingerprint map. With a larger error on fingerprint map reconstruction, the performance of localization can be affected apparently. Even with the reconstructed fingerprint map recovered from the incomplete one, our algorithm still achieves a high accuracy (over 90%) and low ALE (less than 1.5 meters). Therefore, we conclude DeFLoc can realize accurate vehicle localization in indoor parking areas.

VI. Conclusion

In this work, we firstly investigate the reconstruction of FM fingerprint map in an indoor parking environment. By analyzing the feature of FM fingerprints, we propose a deep learning model to do fingerprint map reconstruction with high accuracy. We design smooth layers for both station frequencies and noise frequencies with trainable variables to accelerate the convergence of the network and meanwhile improve the reconstruction performance slightly. We also propose a continuous localization algorithm for precise localization prediction. Field tests in the real indoor parking area demonstrate that our proposed reconstruction model outperforms conventional matrix completion methods, and helps to achieve accurate localization. In our future work, we consider combining FM and other signal sources, such as WiFi, to further improve the accuracy of indoor vehicle localization.

References

[1] M. Khalid, K. Wang, N. Aslam, Y. Cao, N. Ahmad, and M. K. Khan, "From smart parking towards autonomous valet parking: A survey, challenges and future works," *J. Netw. Comput. Appl.*, vol. 175, Feb. 2021, Art. no. 102935.

[2] A. El-Rabbany, *Introduction to GPS: The Global Positioning System*. Norwood, MA, USA: Artech House, 2002.

[3] J. LaMance, J. DeSalas, and J. Jarvinen, "Assisted GPS: A low-infrastructure approach," *GPS World*, vol. 13, no. 3, pp. 46–51, 2002.

[4] H. S. Ramos, T. Zhang, J. Liu, N. B. Priyantha, and A. Kansal, "LEAP: A low energy assisted GPS for trajectory-based services," in *Proc. ACM UbiComp*, 2011, pp. 335–344.

[5] F. Liu *et al.*, "Survey on WiFi-based indoor positioning techniques," *IET Commun.*, vol. 14, no. 9, pp. 1372–1383, 2020.

[6] S. Yoon, K. Lee, and I. Rhee, "FM-based indoor localization via automatic fingerprint DB construction and matching," in *Proc. ACM MobiSys*, Taipei, Taiwan, 2013, p. 207.

[7] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *Proc. ACM MobiSys*. Windermere, U.K.: Low Wood Bay, 2012, p. 169.

[8] X. Chen, Q. Xiang, L. Kong, and X. Liu, "RadioLoc: Learning vehicle locations with FM signal in all-terrain environments," in *Proc. IEEE MASS*, Nov. 2019, pp. 438–446.

[9] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *IEEE INFOCOM*, vol. 2, Mar. 2000, pp. 775–784.

[10] J. J. M. Diaz, R. D. A. Maues, R. B. Soares, E. F. Nakamura, and C. M. S. Figueiredo, "Bluepass: An indoor bluetooth-based localization system for mobile applications," in *Proc. IEEE ISCC*, Jun. 2010, pp. 778–783.

[11] J. Xiong and K. Jamieson, "ArrayTrack: A fine-grained indoor location system," in *Proc. USENIX NSDI*, 2013, pp. 71–84.

[12] J. Xiong, K. Sundaresan, and K. Jamieson, "ToneTrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *Proc. ACM MobiCom*, Paris France, Sep. 2015, pp. 537–549.

[13] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. ACM MobiSys*, Seattle, WA, USA, 2005, p. 205.

[14] Z. Jiang, W. Xi, X.-Y. Li, J. Zhao, and J. Han, "HiLoc: A TDoA-fingerprint hybrid indoor localization system," Microsoft Indoor Localization Competition, Microsoft, Berlin, Germany, Tech. Rep., 2014.

[15] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," in *Proc. ACM IPSN*, Apr. 2015, pp. 178–189.

[16] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "CSI-based indoor localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1300–1309, Jul. 2013.

[17] X. Shen, Y. Chen, J. Zhang, L. Wang, G. Dai, and T. He, "BarFi: Barometer-aided Wi-Fi floor localization using crowdsourcing," in *Proc. IEEE MASS*, Oct. 2015, pp. 416–424.

[18] H. Liu, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Accurate WiFi based localization for smartphones using peer assistance," *IEEE Trans. Mobile Comput.*, vol. 13, no. 10, pp. 2199–2214, Oct. 2014.

[19] M. Chen, K. Liu, J. Ma, Y. Gu, Z. Dong, and C. Liu, "SWIM: Speed-aware WiFi-based passive indoor localization for mobile ship environment," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 765–779, Feb. 2021.

[20] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. ACM Mobicom*. Istanbul, Turkey: ACM Press, 2012, p. 269.

[21] X. Liu *et al.*, "Accurate localization of tagged objects using mobile RFID-augmented robots," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1273–1284, Apr. 2021.

[22] K. Urano, K. Hiroi, T. Yonezawa, and N. Kawaguchi, "Basic study of BLE indoor localization using LSTM-based neural network (poster)," in *Proc. ACM MobiSys*, Jun. 2019, pp. 558–559.

[23] K. Mundnich, B. Girault, and S. Narayanan, "Bluetooth based indoor localization using triplet embeddings," in *Proc. IEEE ICASSP*, May 2019, pp. 7570–7574.

[24] R. Margolies *et al.*, "Can you find me now? Evaluation of network-based localization in a 4G LTE network," in *Proc. IEEE INFOCOM*, May 2017, pp. 1–9.

[25] C. R. Johnson, "Matrix completion problems: A survey," in *Matrix Theory and Applications*, vol. 40. Providence, RI, USA: AMS, 1990, pp. 171–198.

[26] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[27] P. Di Lena, C. Sala, A. Prodi, and C. Nardini, "Missing value estimation methods for DNA methylation data," *Bioinformatics*, vol. 35, no. 19, pp. 3786–3793, Oct. 2019.

[28] K. Cho and N. Reyhani, "An iterative algorithm for singular value decomposition on noisy incomplete matrices," in *Proc. IEEE IJCNN*, Jun. 2012, pp. 1–6.

[29] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *J. Mach. Learn. Res.*, vol. 11, pp. 2287–2322, Jan. 2010.

[30] J. Fan and T. Chow, "Deep learning based matrix completion," *Neurocomputing*, vol. 266, pp. 540–549, Nov. 2017.

[31] J. Fan and J. Cheng, "Matrix completion by deep matrix factorization," *Neural Netw.*, vol. 98, pp. 34–41, Feb. 2018.

[32] X. Han, B. Wu, Z. Shou, X.-Y. Liu, Y. Zhang, and L. Kong, "Tensor FISTA-Net for real-time snapshot compressive imaging," in *Proc. AAAI*, 2020, vol. 34, no. 7, pp. 10933–10940.

[33] P. Yang, L. Kong, X.-Y. Liu, X. Yuan, and G. Chen, "Shearlet enhanced snapshot compressive imaging," *IEEE Trans. Image Process.*, vol. 29, pp. 6466–6481, 2020.

[34] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1654–1662.

[35] Z. Chen, W. Zhao, and S. Wang, "Kernel meets recommender systems: A multi-kernel interpolation for matrix completion," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114436.

[36] H. Li, S. J. Pan, R. Wan, and A. C. Kot, "Heterogeneous transfer learning via deep matrix completion with adversarial kernel embedding," in *Proc. AAAI*, 2019, vol. 33, no. 1, pp. 8602–8609.

[37] X.-Y. Liu, S. Aeron, V. Aggarwal, X. Wang, and M.-Y. Wu, "Adaptive sampling of RF fingerprints for fine-grained indoor localization," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2411–2423, Oct. 2016.

[38] Z. Gu, Z. Chen, Y. Zhang, Y. Zhu, M. Lu, and A. Chen, "Reducing fingerprint collection for indoor localization," *Comput. Commun.*, vol. 83, pp. 56–63, Jun. 2016.

[39] J. Cheng, Z. Song, Q. Ye, and H. Du, "MIL: A mobile indoor localization scheme based on matrix completion," in *Proc. IEEE ICC*, May 2016, pp. 1–5.

[40] K. T. Chui, D. C. L. Fung, M. D. Lytras, and T. M. Lam, "Predicting at-risk university students in a virtual learning environment via a machine learning algorithm," *Comput. Hum. Behav.*, vol. 107, Jun. 2020, Art. no. 105584.

[41] A. Rubinsteyn and S. Feldman. *Fancyimpute: An Imputation Library for Python*. Accessed: 2016. [Online]. Available: https://github.com/iskandr/fancyimpute

Jiale Lei received the B.Eng. degree in computer science and technology from the Shanghai University of Finance and Economics, Shanghai, China, in 2020. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai. His research interests include wireless communication, mobile computing, and machine learning.

Junqin Huang received the B.Eng. degree in computer science and technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include crowdsensing, the Internet of Things, blockchain, and mobile computing.

Linghe Kong (Senior Member, IEEE) received the B.Eng. degree in automation from Xidian University in 2005, the master's degree in telecommunication from Telecom SudParis in 2007, and the Ph.D. degree in computer science from Shanghai Jiao Tong University in 2013. He is currently a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Before that, he was a Post-Doctoral Researcher with Columbia University, McGill University, and the Singapore University of Technology and Design. His research interests include the Internet of Things, 5G, blockchain, and mobile computing.

Guihai Chen received the B.S. degree from Nanjing University in 1984, the M.E. degree from Southeast University in 1987, and the Ph.D. degree from The University of Hong Kong in 1997. He is currently a Distinguished Professor with Shanghai Jiao Tong University, China. He was a Visiting Professor with many universities, including the Kyushu Institute of Technology, Japan, in 1998; The University of Queensland, Australia, in 2000; and Wayne State University, USA, from 2001 to 2003. His research interests include sensor networks, peer-to-peer computing, and high-performance computer architecture and combinatorics.

Muhammad Khurram Khan (Senior Member, IEEE) is currently working as a Professor of cybersecurity with the Center of Excellence in Information Assurance, King Saud University, Saudi Arabia. He is the Founder and the CEO of the Global Foundation for Cyber Studies and Research, Washington, DC, USA, an independent and non-partisan cybersecurity think-tank. He has published more than 400 papers in the journals and conferences of international repute. In addition, he is an Inventor of ten US/PCT patents. He has edited ten books/proceedings published by Springer-Verlag, Taylor and Francis, and IEEE. His research interests include cybersecurity, digital authentication, the IoT security, biometrics, multimedia security, cloud computing security, cyber policy, and technological innovation management. He is a fellow of the IET, U.K.; the BCS, U.K.; and the FTRA, South Korea. He is the Vice Chair of IEEE Communications Society Saudi Chapter. He is a Distinguished Lecturer of the IEEE. He is the Editor-in-Chief of *Telecommunication Systems* (Springer-Nature) with its recent impact factor of 1.73 (JCR 2020). He is on the Editorial Board of several journals, including IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, *IEEE Communications Magazine*, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, *Journal of Network and Computer Applications* (Elsevier), IEEE ACCESS, *IEEE Consumer Electronics Magazine*, *PLOS One*, and *Electronic Commerce Research*.