MetaLoc: Learning to Learn Wireless Localization

Jun Gao*†, Dongze Wu*, Feng Yin*, Qinglei Kong‡, Lexi Xu§, Shuguang Cui*†
*School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China
†Future Network of Intelligence Institute (FNii), The Chinese University of Hong Kong, Shenzhen, China
‡Institute of Space Science and Applied Technology, Harbin Institute of Technology, Shenzhen, China
§Research Institute, China United Network Communications Corporation, Beijing, China

***Abstract*—Existing localization methods that intensively leverage the environment-specific received signal strength (RSS) or channel state information (CSI) of wireless signals are rather accurate in certain environments. However, these methods, whether based on pure statistical signal processing or data-driven approaches, often struggle to generalize to changing environments, which result in significant knowledge and effort losses. To address this challenge, we propose MetaLoc, a fingerprinting-based localization framework that leverages model-agnostic meta-learning (MAML) to achieve fast adaptation to new environments with minimal human intervention. Implemented using a deep neural network with strong representation capabilities, MetaLoc is trained on historical data collected from well-calibrated environments to optimize meta-parameters for fast adaptation. The framework includes two paradigms for obtaining meta-parameters: the centralized paradigm, which optimizes meta-parameters by sharing data from all historical environments with simplicity, and the distributed paradigm, which preserves data privacy by training environment-specific meta-parameters independently. Furthermore, the advanced distributed paradigm modifies the vanilla MAML loss function to ensure that the reduction of loss occurs in a consistent direction across various training domains, thus facilitating faster convergence during training. Our experiments on both synthetic and real datasets demonstrate that MetaLoc outperforms baseline methods in terms of localization accuracy, robustness, and cost-effectiveness. The code and datasets used in this study are publicly available[1].**

***Index Terms*—CSI, meta-learning, RSS, sample efficiency, wireless localization.**

I. Introduction

Location-based services have become an integral part of our daily lives, and various localization techniques have been extensively researched for over a century by various scientific communities [2]–[5]. Although existing global navigation satellite systems (GNSS) provide high outdoor localization accuracy, the demands of emerging applications in diverse areas, such as autonomous driving [6], cooperative 3D scene reconstruction [7], and epidemic tracking [8], require higher levels of accuracy, cost-effectiveness, and robustness. Moreover, the emergence of mmWave and massive MIMO technologies in 5G and 6G communications also demands precise location information for high-throughput transmissions [9]–[11]. In light of these growing needs, it is crucial to develop a state-of-the-art localization system that can cater to both outdoor and complex indoor environments to achieve a full range of high-precision location-based services [12]–[14].

The field of fingerprinting-based localization has recently gained much attention for indoor environments, and the process involves two stages: the offline stage and the online stage. During the offline stage, features of received signals at reference points (RPs) with known locations are collected to form a fingerprint database. These signals are transmitted by access points (APs) in the environment, and the representative features include received signal strength (RSS), channel state information (CSI), and magnetic field information. In the online stage, signal features collected at an unknown location, also known as a test point (TP), are compared with the established database to determine the predicted location using algorithms such as RADAR [15] and Horus [16].

Wireless signal propagation is susceptible to even slight changes in the environment, such as opening a door or the presence of people moving around, which can result in inconsistent fingerprints even if a user remains in the same location. This makes it challenging to construct an accurate statistical fingerprint database that accurately represents the entire area of interest. To tackle this challenge, data-driven localization mechanisms using machine learning techniques have gained increasing attention in recent years [17]–[19]. While machine learning techniques are data-hungry and require large amounts of data samples, building a database for each target indoor localization can be time-consuming and labor-intensive. To overcome this challenge, various established machine learning techniques have been applied to indoor localization, such as data augmentation [20], semi-supervised learning techniques [21], and informed machine learning [22].

However, despite the advances made in indoor localization using machine learning, it is still a challenge to build a model that is universally applicable to all indoor environments. Most studies in the field to date have only focused on one specific environment, such as a single room or floor of a building [15], [23], [24]. This narrow focus means that there is no guarantee that a pre-selected machine learning model that performs well in one environment will be effective in others. To use a machine learning model in a new environment, it is often necessary to collect a large amount of data from site surveys and then rebuild the corresponding fingerprint database and

This paper is an extension of our work [1] initially presented in the proceedings of the IEEE International Conference on Communications (ICC), Seoul, Korea, in May 2022.

Corresponding author: Feng Yin, E-mail: yinfeng@cuhk.edu.cn

[1]Codes and datasets are found at:https://github.com/dongzewuu/MetaLoc

TABLE I
COMPARISONS OF DIFFERENT LOCALIZATION METHODS

Methods	Signal Features	Models	Accuracy	Robustness	Cost-effectiveness
TransLoc [25]	RSS	Machine learning	1.82 m∼2.81 m	✓	×
ViVi [26]	RSS	Deterministic	3.30 m∼4.30 m	✓	×
AcMu [27]	RSS	Deterministic	1.40 m∼3.00 m	✓	×
DFPS [28]	RSS	Machine Learning	1.2m∼2.8m	✓	×
FILA [29]	CSI	Deterministic	0.45 m ∼ 1.2 m	×	×
DeepFi [30]	CSI	Probabilistic	0.95 m∼1.80 m	×	✓
CiFi [31]	CSI	Machine learning	1.50 m∼3.00 m	×	✓
ConFi [32]	CSI	Machine learning	1.36 m	×	×
CRISLoc [33]	CSI	Machine learning	0.29 m	✓	×
Fidora [34]	CSI	Machine Learning	submeter-level	✓	×
DAFI [35]	CSI	Machine Learning	97.6% 89.3%	✓	×
ILCL [36]	CSI	Probabilistic	1.28m∼2.38m	✓	✓

retrain the model, which can be time-consuming and resource-intensive. This highlights the need for a machine learning model that can learn the essential channel features and be broadly applicable to all indoor environments, as outlined in a recent 6G white paper [13].

A. Related Works

Indoor localization techniques have been developed for many years, and the existing methods can be generally summarized from the perspective of classic signal processing (including both probabilistic and deterministic methods) and machine learning-based localization.

1) Probabilistic Localization: Probabilistic localization is a method that utilizes statistical information to determine the location of a target. This is achieved by comparing the received signal measurements with a pre-built fingerprint database. One of the well-known probabilistic localization methods is Horus [16], which employs a probabilistic model to characterize the signal distribution and calculates the maximum posterior probability of the target's location. Another approach, as discussed in [19], involves the use of Bayesian networks for cooperative localization based on RSS. DeepFi [30] further optimizes computational efficiency by combining a probabilistic model with a greedy learning algorithm. Despite having lower computational requirements, probabilistic localization methods can be challenging to implement in dynamic environments as they rely on accurate position-related measurements.

2) Deterministic Localization: The deterministic methods for indoor localization mainly rely on the similarity metric in the signal space to estimate the physical location of a target. This is done by determining the closest fingerprint location in the signal space, which serves as the estimated location of the target. The most widely used deterministic method is the K-nearest neighbors (KNN) algorithm, which considers various similarity measures between the target's signal and the fingerprints in the database. Some commonly used similarity measures include the Euclidean distance [26], its temporal weighted version [37], the cosine similarity [38], the Tanimoto similarity [39], and others. While deterministic localization methods are relatively straightforward to implement, they can be affected by statistical fluctuations in wireless signals, which can result in a dispersed set of neighbors that are far apart in physical space, leading to less accurate localization.

3) Machine learning-based Localization: Recently, machine learning has played a significant role in the field of localization services. Ghzali et al. attempted to address the difficulties associated with indoor localization by approaching it as a regression problem based on RSS gathered in a real office space. Their proposed solution relied on a neural network with random initialization, which unfortunately necessitated the collection of a substantial amount of data in order to train the model effectively [40]. ConFi [32] was the first work to explore the use of convolutional neural networks (CNNs) for learning CSI images at RP, opening up new possibilities for indoor localization. Hsieh et al. attempted to solve the indoor localization challenge by utilizing the RSS and CSI data as a classification problem. They evaluated various neural network architectures in an effort to find the best fit for accurately estimating the location of an object within a specific room [17]. Despite the potential of these machine learning-based methods, the dynamic nature of indoor environments continues to pose challenges to their robustness.

Improved robustness in localization is being achieved through both the model and data sides. On the model side, domain adaptation techniques like transfer learning are widely used, where a source domain is the original environment and the target domain is a new and potentially unseen environment. For example, TransLoc [25] uses transfer learning to find the appropriate cross-domain mappings and create a homogeneous feature space that contains discriminative information from different domains. CRISLoc [33] also uses transfer learning to reconstruct a high-dimensional CSI fingerprint database based on outdated fingerprints and a few new measurements. Fidora [34] trains a domain-adaptive classifier that adjusts itself to new data using a variational autoencoder, as well as a joint classification and reconstruction structure. ILCL [36] uses incremental learning and expands neural nodes for adaptation and reduced training time, but can still overfit with small numbers of CSI images. On the data side, ViVi [26] reduces uncertainty in RSS fingerprints by exploiting spatial gradients among multiple locations. CiFi [31] uses phase differences between antenna pairs instead of raw measurements to improve the stability of CSI fingerprints. DFPS [28] combines raw RSS and the difference between AP pairs to enhance robustness against heterogeneous hardware.

Despite efforts to find cost-effective alternatives for tra-

TABLE II
IMPORTANT NOTATIONS USED THROUGHOUT THE PAPER

Notation	Description
$\mathcal{P}(\tau)$	The overall distribution of tasks
$\mathcal{P}_i(\tau)$	Distribution of tasks from domain i
$f_{\boldsymbol{\theta}}$	Model output with parameters $\boldsymbol{\theta}$
N	The number of classes in each task
k_{spt}	The number of samples of the support set in each task
k_{qry}	The number of samples of the query set in each task
$D^s_{\tau_i}$	Support set of localization task τ_i that contains k_{spt} number of samples under each location of N ways
$D^q_{\tau_i}$	Query set of localization task τ_i that contains k_{qry} number of samples under each location of N ways
$\mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}, D^s_{\tau_i})$	The task-specific loss function for task τ_i based on model parameters $\boldsymbol{\theta}$ and support set $D^s_{\tau_i}$
$\mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}, D^q_{\tau_i})$	The task-specific loss function for task τ_i based on model parameters $\boldsymbol{\theta}$ and query set $D^q_{\tau_i}$
$\boldsymbol{\theta}'_i$	The task-specific parameters after the inner loop via one step of gradient descent
$\boldsymbol{\theta}^*$	Updated meta-parameters after the outer loop
$\boldsymbol{\theta}_T(Q)$	The task-specific adapted model parameters obtained by updating $\boldsymbol{\theta}^*$ after Q steps of gradient descent
α	The step size of the inner loop
β	The step size of the outer loop

ditional site surveys in the realm of indoor localization, the problem remains persistent. The semi-supervised learning technique [21], which incorporates a limited number of labeled data samples with an abundant amount of low-cost unlabeled data, has been proposed as a potential solution. However, it still struggles to mitigate the challenge of learning the noise information in the data rather than the true patterns. The informed machine learning approach [22], [41], which integrates data and prior knowledge, is another effort to enrich the information contained in the training data, but the use of computer simulation results as a knowledge representation can only do so much. Meanwhile, crowdsourcing and federated learning-based approaches [42], [43] allow for decomposing large-scale fingerprint collections into smaller and local tasks, enabling mobile users to participate using heterogeneous devices, but they struggle to deal with the inconsistencies of fingerprints over space and time. The difficulties in overcoming these challenges highlight the ongoing need to find a cost-effective indoor fingerprinting-based localization method with fewer site surveys.

Table I provides a comparison of various indoor localization methods in terms of accuracy, robustness, and cost-effectiveness. However, it must be acknowledged that the reported localization accuracy in different studies cannot be fairly compared due to the disparate datasets utilized.

B. Contributions

The paper introduces a pioneering localization framework called *MetaLoc*, which leverages the power of meta-learning to revolutionize fingerprinting-based indoor localization. It comprises of two paradigms: (1) a centralized paradigm based on the vanilla model-agnostic meta-learning (MAML) [44] and (2) a distributed paradigm combining MAML with task similarity (MAML-TS) and MAML with domain generalization (MAML-DG). MetaLoc stands out from the crowd by quickly and efficiently adapting to new environments through a small number of newly collected measurements. The framework uses the meta-parameters learned from historical environments to initialize a neural network, breaking through the traditional environment-specific localization bottleneck.

The main contributions of this work are fourfold:

1) **Outstanding Localization Performance**: MetaLoc is the first work to harness the potential of meta-learning for wireless localization. Its ability to quickly adapt to environmental changes with computationally inexpensive updates sets it apart from the competition. Furthermore, MetaLoc achieves this adaptation with just three CSI images per point in a new environment, solidifying its position as a cost-effective solution.
2) **Comprehensive Real-World Dataset:** To fairly evaluate the performance of different localization methods, we have created a publicly available dataset. Our dataset was built using a smartphone as the receiver and three different types of WiFi routers as the transmitters. We conducted field trials on two different scenarios: a spacious hall with minimal obstacles and a cluttered lab. To ensure comprehensive data collection, we took five independent measurements over different days, each at 90 grid points. This resulted in 260 CSI images and corresponding RSS vectors per grid point, providing a rich source of data for comparison.
3) **Innovative Proposed Paradigms**: We propose two paradigms for training meta-parameters in wireless localization. Our first paradigm, a centralized approach, trains the meta-parameters using all historical environments, resulting in a simplified implementation process. Our second paradigm, a distributed approach, protects the privacy of each environment by training environment-specific meta-parameters. The extensive theoretical analyses and experimental results provide evidence of the superiority of our innovative paradigms.
4) **Flexible Model Compatibility**: The work sets its sights on indoor fingerprinting-based localization that leverages both RSS and CSI wireless signal features. Essentially, MetaLoc with its remarkable compatibility, can be utilized with any model that has been trained through gradient descent, and any wireless signal features that can be input into a learning model. This compatibility makes it a highly valuable tool for solving various data-driven wireless localization challenges, including

both regression and classification problems, and holds tremendous potential for the future of wireless localization.

The remainder of the paper is structured as follows: Section II gives the preliminaries of indoor localization. The proposed MetaLoc is then presented in detail in Section III. Section IV outlines the experimental setup. In Section V, we demonstrate the efficacy of the proposed scheme through simulation and real-world data, while the paper is concluded in Section VI. For clarity, the notations adopted throughout the paper are summarized in Table II.

II. Preliminaries

To validate the proposed framework, we focus on the two widely-used wireless signal features for indoor localization: RSS and CSI. A comprehensive overview of fingerprinting-based localization is also provided in this section.

A. Received Signal Strength (RSS)

RSS is a metric in decibels that measures the strength of the radio signal during propagation. It can be easily obtained in various real-world wireless networks without the need for additional infrastructure. The widely adopted path-loss model captures the signal attenuation in indoor environments at the GHz frequency band as follows [45]:

$$P_t - P_r = 10n\lg(d) - G + 20\lg\left(\frac{4\pi}{\lambda}\right) + X_\sigma, \qquad (1)$$

where P_t represents the transmit power of the APs, and P_r is the RSS at a RP that is located d meters away from the transmitter. The path-loss exponent n accounts for the impact of obstacles such as walls and doors, whose value is typically greater than two in buildings with blocked paths. G represents the antenna gain, and λ is the wavelength of the wireless signal. X_α is a normal random variable with a standard deviation (std) of σ, which ranges from 3 dB to 20 dB.

Although the vanilla path-loss model captures the impact of distance and environment on the signal strength, it is not always able to precisely describe the complex signal attenuation, as the RSS is susceptible to changes in the real environment.

B. Channel State Information (CSI)

Orthogonal frequency division multiplexing (OFDM) is a widely used technology in the latest wireless communication standards, including 802.11a, 802.11n, and 802.11ac. The CSI extracted from the OFDM receivers can reveal the multipath characteristics of the wireless channel. In general, CSI is defined as the ratio of received signal to transmitted signal and is represented by a complex number, $H_i = |H_i|e^{j\sin(\angle H_i)}$, where $|H_i|$ and $\angle H_i$ represent the amplitude and phase of the i-th subcarrier, respectively.

While the commonly used CSI collection toolkit, Intel 5300 CSI tool [46], requires extensive hardware support and a successful connection to each AP, we adopt a more convenient and accessible option with Nexmon [47], a smartphone-based CSI collection tool that does not require connection to surrounding encrypted APs.

In conclusion, CSI is a detailed and sophisticated measurement in the frequency domain compared with the convenient RSS. With multiple subcarriers providing information about different fading or scattered paths, CSI allows for the construction of robust fingerprints that characterize each location and enable the design of accurate localization systems.

C. Fingerprinting-based Localization

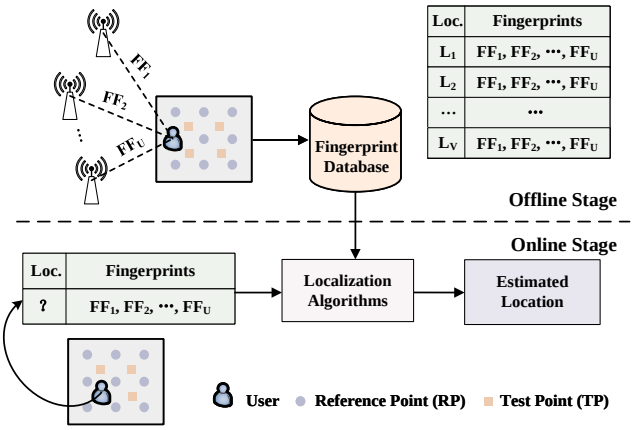


Fig. 1. A diagram of fingerprinting-based localization, where FF represents the fingerprint fragment collected from each AP, which can be RSS or CSI. Loc. represents the location information.

Figure 1 illustrates the diagram of fingerprinting-based localization. The process is divided into two stages: the offline stage and the online stage. In the offline stage, a database includes fingerprints and the corresponding location information is created. This involves deploying multiple APs as transmitters, and a user with a mobile device as the receiver. The signal features collected from each AP, such as received RSS or CSI, form a fingerprint fragment (FF). When multiple FFs are received from U APs, they are combined to create a fingerprint that characterizes a specific location L_i ($i = 1, 2, \ldots, V$). During the online stage, fingerprints are collected at TPs to estimate the location using localization algorithms. Next, we will delve into our proposed fingerprint

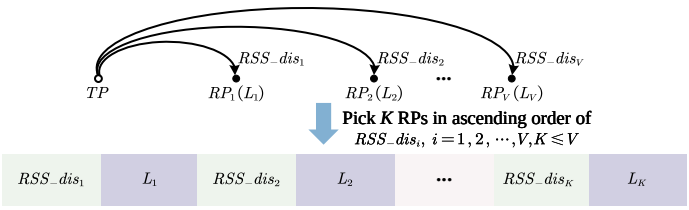
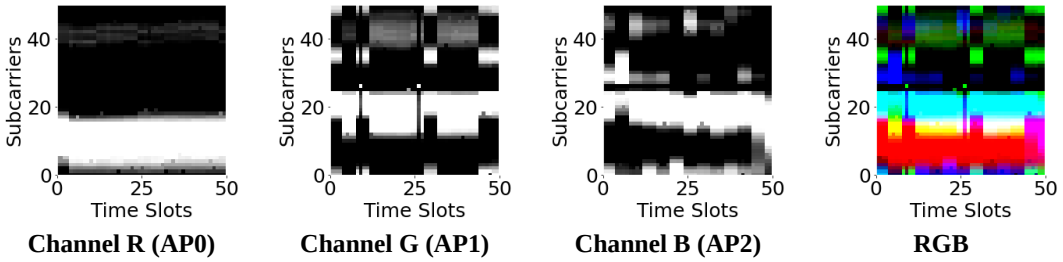


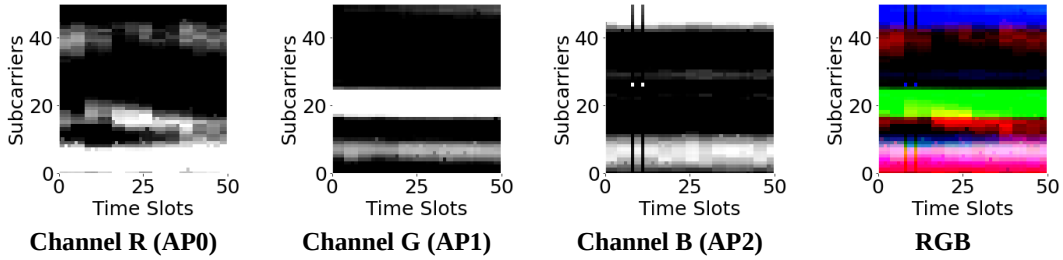
Fig. 2. The designed RSS fingerprints.

design that uses both RSS and CSI.

1) RSS: To determine the location of each TP in our fingerprint database, we utilize the estimated TP's K-closest RPs in the signal space. This information includes the K-nearest Euclidean distances and the physical coordinates of



(a) CSI fingerprints for different channels at location 1.



(b) CSI fingerprints for different channels at location 2.

Fig. 3. The newly designed CSI fingerprints.

the corresponding RPs, as depicted in Fig. 2. In particular, RSS_dis_i represents the i-th closest Euclidean distance between the estimated TP and all the RPs in the signal space, while L_i denotes the corresponding location of the i-th RP.

2) CSI: Nexmon has the ability to extract 52 valuable subcarriers from each AP, allowing us to create CSI images with amplitude information for each location, as seen in Fig. 3. The CSI images, with dimensions $52\times50\times3$, consist of 50 data packets (rows) and 52 measured amplitude values (columns) from subcarriers. The three channels of the CSI image represent information received from the three APs deployed in the environment, similar to the RGB channels in a colorful image. The CSI images highlight distinctive characteristics between two separate locations as shown in Fig. 3, making them ideal candidates as fingerprints for localization.

Moreover, we use a histogram intersection method to assess the similarity of two different CSI images. Given an image, we divide all its pixels into equal-interval bins and create a histogram, where each bar represents the pixel count in that bin. Let $Z_j(I)$ and $Z_j(I')$ denote the pixel count in the j-th bin of images I and I', respectively. The histogram intersection of two images each with n bins is defined as [48]:

$$Z(I) \cap Z\left(I'\right) = \sum_{j=1}^{n} \min\left(Z_j(I), Z_j\left(I'\right)\right), \tag{2}$$

where $\min(x, y)$ function takes two values x and y as arguments and returns the smaller one. The histogram intersection calculates the similarity between the two images by summing up the minimum number of pixels in each bin of both images. In other words, it measures the overlap between the two histograms. A larger histogram intersection value indicates that the two images are more similar. In the following experiments, $n = 256$ bins are used for the designed CSI images with pixel values ranging from 0 to 255.

III. The Proposed System

MetaLoc is an innovative solution that leverages the power of neural networks to perform localization tasks in new environments with remarkable efficiency and accuracy. Fig. 4 presents a clear picture of the challenges and the proposed MetaLoc framework, which provides a unique and effective approach to tackle the changing nature of environments.

In a transmitter-receiver deployment scenario, the main challenge is that changes in the environment, such as the movement of obstacles, can significantly impact the propagation conditions and result in alterations to the fingerprints of the same location point. This can render the existing fingerprint database outdated and in need of an update to train a new neural network. We refer to previous environments where calibration was done as historical environments and to the environments where localization is to be performed as new environments.

To tackle this challenge head-on, the proposed MetaLoc is divided into two stages: meta-training and meta-test. The meta-training stage requires data gathered from historical environments, while the meta-test stage only necessitates a minimal amount of data in the new environment. The meta-parameters $\boldsymbol{\theta}^*$ learned from the meta-training stage provide a strong starting point for the training process in the meta-test stage, eliminating the need to start from scratch in the face of environmental variations.

A. Meta-learning

Meta-learning is a learning to learn approach that enables the learning model to adapt to new tasks by leveraging previous experience from related tasks. In this framework, tasks are drawn from a specific distribution, denoted as $\tau \sim \mathcal{P}(\tau)$, and each task includes a support set for training and a query set for test. In an N-way k-shot classification problem, a task

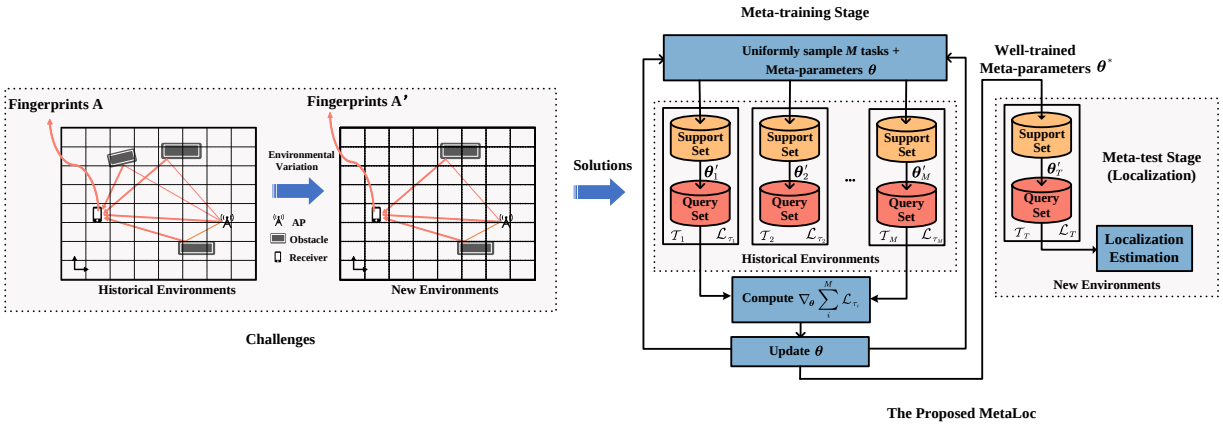


Fig. 4. The overview of the proposed MetaLoc framework.

consists of N classes, each with k samples. In the meta-training stage, M training tasks, $\{\tau_i\}_{i=1}^{M} \sim \mathcal{P}(\tau)$, are sampled from the distribution and the corresponding datasets are made available to the model. In the meta-testing stage, a new test task $T \sim \mathcal{P}(\tau)$ is presented, consisting of a small support set and a query set. The objective of meta-learning is to train a model on the M training tasks, such that it can quickly adapt to the new test task using the small support set and perform well on the query set.

Model-agnostic meta-learning (MAML) does so by learning a set of initial parameters $\boldsymbol{\theta}_{MAML}$ for neural networks that enable good performance on a new task with only a few gradient descent steps. In the meta-training stage, MAML formulates a meta-optimization problem to find $\boldsymbol{\theta}_{MAML}$ as:

$$\boldsymbol{\theta}_{MAML} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{M} \mathcal{L}_{\tau_i}\left(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \hat{\mathcal{L}}_{\tau_i}(\boldsymbol{\theta})\right), \quad (3)$$

where it contains two task-specific loss functions $\hat{\mathcal{L}}_{\tau_i}$ and $\mathcal{L}_{\tau_i}$ computed based on the support set and query set of the training task τ_i, respectively. Then the meta-parameters are updated via stochastic gradient descent (SGD):

$$\boldsymbol{\theta}_{MAML} \leftarrow \boldsymbol{\theta}_{MAML} - \beta \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{M} \mathcal{L}_{\tau_i}\left(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \hat{\mathcal{L}}_{\tau_i}(\boldsymbol{\theta})\right), \quad (4)$$

where α and β denote the step size of the inner loop and outer loop, respectively. During the meta-test stage, the meta-parameters $\boldsymbol{\theta}_{MAML}$ are fine-tuned to obtain the parameters $\boldsymbol{\theta}_T$ for the neural network used in the test task T. This is achieved by updating the meta-parameters using the gradient of the loss function $\hat{\mathcal{L}}_T(\boldsymbol{\theta}_{MAML})$ computed based on the support set of the test task, as follows:

$$\boldsymbol{\theta}_T \leftarrow \boldsymbol{\theta}_{MAML} - \alpha \nabla_{\boldsymbol{\theta}} \hat{\mathcal{L}}_T(\boldsymbol{\theta}_{MAML}), \quad (5)$$

In the following, we introduce two paradigms for the implementation of MetaLoc to learn the meta-parameters, namely the centralized paradigm and the distributed paradigm. The centralized paradigm shares data collected from all historical environments to derive the meta-parameters, while the distributed paradigm maintains the privacy of data in each environment by training environment-specific meta-parameters separately. The advanced distributed paradigm of MetaLoc overcomes the domain shift challenge by leveraging the exchange of environment-specific meta-parameters between different environments.

B. Centralized MetaLoc Paradigm

The centralized paradigm of MetaLoc is built upon the foundation of the vanilla MAML algorithm [44]. In this paradigm, the data gathered from various historical environments as a whole is employed during the meta-training stage to learn the meta-parameters.

1) Meta-training stage: The primary purpose of the meta-training stage is to derive the well-trained meta-parameters $\boldsymbol{\theta}^*$. This is achieved through the use of a neural network that maps the observed fingerprints of the estimated TPs to the desired location outputs. The process is carried out in the following steps:

Step ①: The neural network is initialized with a preselected network architecture and a set of randomly initialized meta-parameters $\boldsymbol{\theta}^*$, which are represented by the parameterized function $f_{\boldsymbol{\theta}}$. A total of M tasks $\tau_1, \ldots, \tau_M$ are sampled from the localization task distribution $p(\tau)$ to be used as the training tasks. Each task τ_i is composed of a loss function $\mathcal{L}_{\tau_i}$, a support set $D^s_{\tau_i}$ and a query set $D^q_{\tau_i}$. The loss function $\mathcal{L}_{\tau_i}$ provides task-specific feedback, with cross-entropy being used for classification tasks and either mean-squared-error (MSE) or root-mean-square-error (RMSE) being used for regression tasks.

Step ②: For localization task τ_i, we train $f_{\boldsymbol{\theta}}$ with the support set $D^s_{\tau_i}$ and derive the task-specific loss $\mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}; D^s_{\tau_i})$. We obtain task-specific parameters $\boldsymbol{\theta}'_i$ during the inner loop using one-step gradient descent update, expressed as

$$\boldsymbol{\theta}'_i = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}; D^s_{\tau_i}), \quad (6)$$

where the hyper-parameter α represents the step size of the inner loop. The task-specific parameters $\boldsymbol{\theta}'_i$ obtained in Step ② provide only limited information about each localization task, as they are derived using a one-step gradient descent update on the support set $D^s_{\tau_i}$. Further evaluation of their performance on the query set $D^q_{\tau_i}$ is necessary to obtain a more comprehensive understanding of each task.

Step ③: The performance of one-step update for the i-th localization task can be further evaluated using the samples in the query set given by $\mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q)$. We define the meta-loss as the sum of all task-specific losses, $\sum_{i=1}^{M} \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q)$. This meta-loss is also called the meta-objective. We then update the meta-parameters $\boldsymbol{\theta}$ by minimizing the meta-loss. In this way, the meta-optimization across all M tasks is given by

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{M} \mathcal{L}_{\tau_i}\left(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q\right). \tag{7}$$

It is important to note that during the meta-optimization process, the optimization is performed over the meta-parameters $\boldsymbol{\theta}$, while the meta-objective is computed using the task-specific parameters $\boldsymbol{\theta}_i'$. This separation ensures that the meta-parameters are updated to achieve the best generalization performance across all tasks, while the task-specific parameters are used to evaluate the one-step update performance for each individual task.

Step ④: The meta-optimization during outer loop is carried out using SGD, where the meta-parameters $\boldsymbol{\theta}$ are updated as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{M} \mathcal{L}_{\tau_i}\left(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q\right), \tag{8}$$

where β is the step size for the outer loop. During this optimization, the meta-objective is computed using the adapted task-specific parameters $\boldsymbol{\theta}_i'$. To compute the partial derivative of $\mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q)$ with respect to the m-th element of $\boldsymbol{\theta}$, we can use the chain rule and write:

$$\frac{\partial \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q)}{\partial \boldsymbol{\theta}^{(m)}} = \sum_{n} \frac{\partial \mathcal{L}_{\tau_i}\left(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q\right)}{\partial \boldsymbol{\theta}_i'^{(n)}} \frac{\partial \boldsymbol{\theta}_i'^{(n)}}{\partial \boldsymbol{\theta}^{(m)}}, \tag{9}$$

and by merging Eq. (6), we have

$$\frac{\partial \boldsymbol{\theta}_i'^{(n)}}{\partial \boldsymbol{\theta}^{(m)}} = \frac{\partial \boldsymbol{\theta}^{(n)}}{\partial \boldsymbol{\theta}^{(m)}} - \alpha \frac{\partial \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}; D_{\tau_i}^s)}{\partial \boldsymbol{\theta}^{(n)} \partial \boldsymbol{\theta}^{(m)}}. \tag{10}$$

However, directly computing the partial derivative $\frac{\partial \boldsymbol{\theta}_i'^{(n)}}{\partial \boldsymbol{\theta}^{(m)}}$ using the inner loop update rule involves computing second-order derivatives, which can be computationally expensive. Therefore, we make use of the first-order approximation [49], assuming that the second-order derivatives are negligible. Specifically, we approximate $\frac{\partial \boldsymbol{\theta}_i'^{(n)}}{\partial \boldsymbol{\theta}^{(m)}}$ as follows:
1) When $m \neq n$:

$$\frac{\partial \boldsymbol{\theta}_i'^{(n)}}{\partial \boldsymbol{\theta}^{(m)}} = -\alpha \frac{\partial \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}; D_i^s)}{\partial \boldsymbol{\theta}^{(n)} \partial \boldsymbol{\theta}^{(m)}} \approx 0. \tag{11}$$

2) When $m = n$:

$$\frac{\partial \boldsymbol{\theta}_i'^{(n)}}{\partial \boldsymbol{\theta}^{(m)}} = 1 - \alpha \frac{\partial \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}; D_{\tau_i}^s)}{\partial \boldsymbol{\theta}^{(n)} \partial \boldsymbol{\theta}^{(m)}} \approx 1. \tag{12}$$

Using these approximations, Eq. (9) can be simplified to:

$$\frac{\partial \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q)}{\partial \boldsymbol{\theta}^{(m)}} \approx \frac{\partial \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q)}{\partial \boldsymbol{\theta}_i'^{(m)}}. \tag{13}$$

Therefore, the meta-optimization over the meta-parameters $\boldsymbol{\theta}$ can be simplified to optimization over the task-specific parameters $\boldsymbol{\theta}_i'$, which eliminates the need for complicated second-order derivatives.

2) Meta-test Stage: After the meta-training stage, we obtain well-trained meta-parameters $\boldsymbol{\theta}^*$ that capture the essential characteristics of localization from historical environments. To perform an unseen task in the new environment $T \sim \mathcal{P}(\tau)$ with support set D_T^s and query set D_T^q, we use the pre-trained neural network $f_{\boldsymbol{\theta}^*}$ with a predefined architecture and learned meta-parameters $\boldsymbol{\theta}^*$ as initialization. Then, we update task-specific adapted parameters $\boldsymbol{\theta}_T(Q)$ by taking Q-steps of gradient descent on the small support set D_T^s, which is given by:

$$\boldsymbol{\theta}_T(Q) = \boldsymbol{\theta}^* - \alpha \left[\nabla_{\boldsymbol{\theta}^*} \mathcal{L}_T\left(f_{\boldsymbol{\theta}^*}; D_T^s\right) + \sum_{j=1}^{Q-1} \nabla_{\boldsymbol{\theta}_T(j)} \mathcal{L}_T\left(f_{\boldsymbol{\theta}_T(j)}; D_T^s\right) \right], \tag{14}$$

and task-specific optimal parameters $\boldsymbol{\theta}_T^*$ for test task T is given by

$$\boldsymbol{\theta}_T^* = \arg\min_{\boldsymbol{\theta}_T(Q)} \mathcal{L}_T\left(f_{\boldsymbol{\theta}_T(Q)}; D_T^s\right), \tag{15}$$

where $\mathcal{L}_T\left(f_{\boldsymbol{\theta}_T(Q)}; D_T^s\right)$ denotes the test loss measured on the query set D_T^s. The entire process is summarized in Algorithm 1.

Algorithm 1 Vanilla MAML

Require:
$\mathcal{P}(\tau)$: distribution over tasks;
α: step size of the inner loop;
β: step size of the outer loop;
Meta-training Stage (in the historical environments):
1: Randomly initialize $\boldsymbol{\theta}$;
2: For ite in iterations do:
3: Sample training tasks $\{\tau_i\}_{i=1}^{M} \sim \mathcal{P}(\tau)$;
4: For each i in $\{1, 2, \ldots, M\}$ do:
5: $\boldsymbol{\theta}_i' = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}; D_{\tau_i}^s)$;
6: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{\tau_i} \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q)$;
7: **return** $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$ when it converges.
Meta-test Stage (in the new environment):
8: Sample a test task $T \sim \mathcal{P}(\tau)$;
9: $\boldsymbol{\theta}_T \leftarrow \boldsymbol{\theta}^* - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_T(f_{\boldsymbol{\theta}^*}; D_T^s)$;
10: **return** $\boldsymbol{\theta}_T^* \leftarrow \boldsymbol{\theta}_T$ when it converges.

3) Performance analysis of the vanilla MAML: To further explore the performance of MetaLoc, we borrow the idea of reference [50] to address the following questions: (1) what is the reason for the superior performance of vanilla MAML over traditionally direct training a neural network with a variety of different environments? (2) which factors influence the test performance of vanilla MAML?

Definition (Lipschitz Continuity). A function $f(\boldsymbol{\theta})$ *is said to be G-Lipschitz continuous over a region D (bounded or unbounded) if there exists a G>0 such that* $||f(\boldsymbol{\theta}_1) - f(\boldsymbol{\theta}_2)||_2 \leq G||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2||_2$ *for all* $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in D$. *Moreover,* $f(\boldsymbol{\theta})$ *is said to be* W*-smooth if* $||\nabla f(\boldsymbol{\theta}_1) - \nabla f(\boldsymbol{\theta}_2)||_2 \leq W||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2||_2$.

Next, we define the excess risk with respect to $\boldsymbol{\theta}_T(Q)$ as $ER(\boldsymbol{\theta}_T(Q)) = E_{T\sim\tau} E_{D_T}[\mathcal{L}(\boldsymbol{\theta}_T(Q)) - \mathcal{L}(\boldsymbol{\theta}_T^*)]$, which evaluates the loss difference on all samples from all tasks and well measures the test performance of the adapted parameters $\boldsymbol{\theta}_T(Q)$ over Q gradient steps. The lower the $ER(\boldsymbol{\theta}_T(Q))$, the better adaptation ability the MAML possesses.

Theorem 1. *Suppose $\mathcal{L}_T(f_{\boldsymbol{\theta}}, D_T^s)$ is G-Lipschitz continuous and W-smooth with respect to the parameters $\boldsymbol{\theta}$, and α satisfies $\alpha \leq \frac{1}{W}$. Setting $\rho = 1+2\alpha W$, then for any $T \sim \mathcal{P}(\tau)$ with $D_T^s = \{(x_i, y_i)\}_{i=1}^{k_{spt}} \sim T$, we have*

$$ER(\boldsymbol{\theta}_T^Q) \leq \frac{2G^2(\rho^Q - 1)}{k_{spt} * W} + E_{T\sim\tau} E_{D_T^s}[\mathcal{L}_T(\boldsymbol{\theta}_T^Q; D_T^s) - \mathcal{L}_T(\boldsymbol{\theta}_T^*)]$$
$$\leq \frac{2G^2(\rho^Q - 1)}{k_{spt} * W} + \frac{1}{2\alpha} E_{T\sim\tau}[||\boldsymbol{\theta}^* - \boldsymbol{\theta}_T^*||_2^2].$$

The second inequality implies that a smaller expected distance between $\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}_T^*$ over T (i.e., $E_{T\sim\tau}[||\boldsymbol{\theta}^* - \boldsymbol{\theta}_T^*||_2^2]$) leads to smaller $ER(\boldsymbol{\theta}_T(Q))$. Following this idea, we compare the vanilla MAML with a conventional neural network in the next.

The vanilla MAML trains the meta-parameters $\boldsymbol{\theta}$ as shown in Fig. 5, which demonstrates the paths in the parameters space with M tasks. An inner loop is first conducted based on the support set of each task and obtains the task-specific parameters $\boldsymbol{\theta}_i', i = 1, \ldots, M$. Next, an outer loop is implemented to find each task's most potential direction toward the optimal parameters based on the query set. Finally, we get the direction toward the optimal parameters for each training task (in different colors), and update meta-parameters $\boldsymbol{\theta}$ based on the average across the training tasks (path in black). Through this approach, MAML updates the meta-parameters $\boldsymbol{\theta}$ in a direction that aligns with all training tasks, with each $\boldsymbol{\theta}_i^*$ receiving equal weight in the gradient descents of the outer loop. Consequently, it is expected that $E_{T\sim\tau}[||\boldsymbol{\theta}^* - \boldsymbol{\theta}_T^*||_2^2]$ is smaller in MAML than in other traditional training methods.

It should be noted that conventional neural networks do not employ the terminologies of tasks, inner loops, and outer loops. Instead, models are trained using data from a particular dataset at one time. The issue with this approach is that the optimizer may overfit a single environment by finding a path that achieves rapid loss reduction for that environment but shows slow convergence for other environments. In such cases, $E_{T\sim\tau}[||\boldsymbol{\theta}^* - \boldsymbol{\theta}_T^*||_2^2]$ would be larger than in MAML, as demonstrated in Section V.A.

The simplicity of the centralized paradigm in MetaLoc comes at the cost of potential data privacy issues, as the data required to train the meta-parameters under this paradigm is shared across different environments. Additionally, vanilla MAML fails to consider the relationship between any two historical environments, which is particularly problematic when the training data comes from different measuring times and environments, as the signal features may vary significantly. To address these issues, we propose a distributed paradigm for MetaLoc that allows for better privacy protection and more effective knowledge transfer between related environments.

C. Distributed MetaLoc Paradigm

Compared to the centralized paradigm, which is trained on all historical environments, our previous work [1] proposes an environment-specific meta-parameter approach. This involves clustering the historical environments based on their propagation models and training a unique set of meta-parameters for each group. Instead of using the traditional meta-parameters

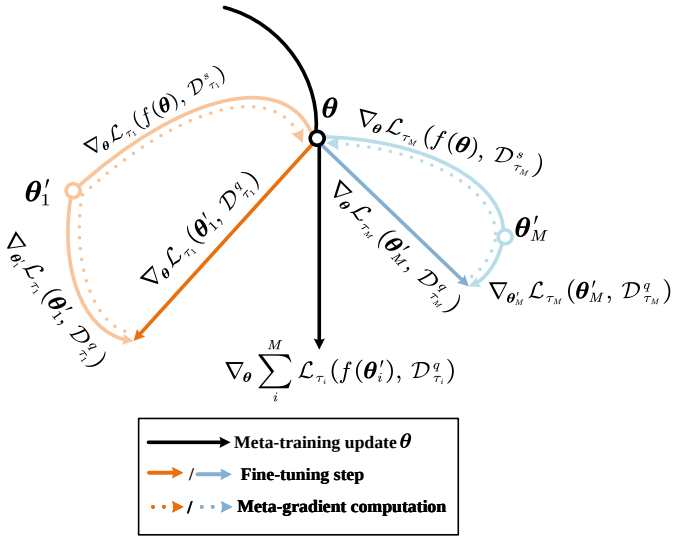


Fig. 5. Diagram of MAML, which optimizes the meta-parameters $\boldsymbol{\theta}$ to obtain the fast adaptation to new tasks.

trained on all tasks, we select the best environment-specific meta-parameters for the target localization task based on task similarity between the test task and the historical environments. To measure this task similarity concretely, we utilize the maximum mean discrepancy (MMD) measure [51], as shown in

$$\mathrm{MMD}[\mathcal{G}, \tau_1, \tau_2] := \sup_{h\in\mathcal{H}} \left(\mathbf{E}_x[h(x)] - \mathbf{E}_y[h(y)]\right), \tag{16}$$

where x and y are wireless signal features, such as RSS fingerprints or CSI fingerprints, for a pair of localization tasks τ_1 and τ_2 in the tasks distribution $\mathcal{P}(\tau)$, respectively. Moreover, $\mathcal{H}$ refers to a class of functions h. In [51], it was shown that when $\mathcal{H}$ is a unit ball in a universal reproducing kernel Hilbert space defined on $\mathcal{P}(\tau)$ with the associated continuous kernel, then $\mathrm{MMD}[\mathcal{H}, \tau_1, \tau_2] = 0$ if and only if τ_1 is equivalent to τ_2. The maximum mean discrepancy (MMD) measure, which quantifies the similarity between tasks, increases with the difference between x and y. Thus, a higher MMD indicates a larger difference between the two tasks.

Since the algorithm considers the task similarity, we name it model-agnostic meta-learning with task similarity (MAML-TS), which allows training in isolation for each environment, thereby ensuring better data security. In our previous work, MAML-TS showed promising performance on simulation data. However, we observe that as the amount of training data decreases during site surveys, it may lead to overfitting. This could be because the environment-specific meta-parameters are trained only on a single environment, which may result in weak generalization ability in complex real-world environments.

To address this issue, we propose model-agnostic meta-learning with domain generalization (MAML-DG), Unlike the MAML-TS approach which trains environment-specific meta-parameters in isolation for each environment, MAML-DG allows the environments to share the environment-specific meta-parameters, while still protecting the raw data of each

environment. This is inspired by the idea of imitating real-time train-test domain shifts [52] to enable the model to quickly generalize to different domains. In the indoor localization setting, we treat each environment as an individual domain, with the historical environments serving as the training domains and the new environment as the test domain.

Algorithm 2 MAML-DG

Require:
$\{\mathcal{P}^{(i)}(\tau)\}_{i=1}^{S}$: distributions over tasks in S domains;
α: step size of the inner loop;
β: step size of the outer loop;
w: weight of the loss function of the second training domain D_{II};
Meta-training Stage (in the historical environments):
1: Randomly initialize $\boldsymbol{\theta}$;
2: For ite in iterations do:
3: Sample two training domains D_I and D_{II} uniformly from $\{1, 2, ..., S\}$;
4: Sample tasks $\{\tau_i^{(D_I)}\}_{i=1}^{M} \sim \mathcal{P}^{(D_I)}(\tau)$ in domain D_I;
5: For i in range (M) do:
6: $\boldsymbol{\theta}_i^{(D_I)} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\tau_i^{(D_I)}}(f_{\boldsymbol{\theta}}; D^s_{\tau_i^{(I)}})$;
7: $\boldsymbol{\theta}' = \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{\tau_i^{(D_I)}} \mathcal{L}_{\tau_i^{(D_I)}}(f_{\boldsymbol{\theta}_i^{(D_I)}}; D^q_{\tau_i^{(D_I)}})$;
8: Sample tasks $\{\tau_j^{(D_{II})}\}_{j=1}^{M} \sim \mathcal{P}^{(D_{II})}(\tau)$ in D_{II};
9: For j in range (M) do:
10: $\boldsymbol{\theta}_j^{(D_{II})} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\tau_j^{(D_{II})}}(f_{\boldsymbol{\theta}'}; D^s_{\tau_j^{(D_{II})}})$;
11: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}' - w\beta \nabla_{\boldsymbol{\theta}} \sum_{\tau_j^{(D_{II})}} \mathcal{L}_{\tau_j^{(D_{II})}}(f_{\boldsymbol{\theta}_j^{(D_{II})}}; D^q_{\tau_j^{(D_{II})}})$;
12: **return** $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$ when it converges.
Meta-test Stage (in the new environment):
13: Sample a test task $T \sim \mathcal{P}(\tau)$;
14: $\boldsymbol{\theta}_T \leftarrow \boldsymbol{\theta}^* - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_T(f_{\boldsymbol{\theta}^*}; D^s_T)$;
15: **return** $\boldsymbol{\theta}^*_T \leftarrow \boldsymbol{\theta}_T$ when it converges.

As outlined in Algorithm 2, MAML-DG is designed to train a deep learning model with parameters $\boldsymbol{\theta}$ across S training domains, which may have different statistical distributions but share the same label and input features space. During each meta-training iteration, MAML-DG randomly selects two training domains $D_I, D_I = 1, 2, \ldots, S$ and $D_{II}, D_{II} = 1, 2, \ldots, S, D_I \neq D_{II}$ and generates tasks in these two domains. The full steps are as follows.

Step ①: We virtually train a domain-specific meta-parameters $\boldsymbol{\theta}'$ on the tasks generated from the training domain D_I using the vanilla MAML algorithm. We derive the first domain-specific loss function as

$$F(\cdot) = \sum_{\tau_i^{(D_I)}} \mathcal{L}_{\tau_i^{(D_I)}}\left(f_{\boldsymbol{\theta}_i^{(D_I)}}\right) = \sum_{i=1}^{M} \mathcal{L}_{\tau_i^{(D_I)}}\left(f_{\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\tau_i^{(D_I)}}}\right). \tag{17}$$

Step ②: With the initialization $\boldsymbol{\theta}'$ obtained in the previous step, we derive a second domain-specific loss function for the tasks generated from the training domain D_{II} using vanilla MAML once again, which is shown as

$$G(\cdot) = \sum_{\tau_j^{(D_{II})}} \mathcal{L}_{\tau_j^{(D_{II})}}(f_{\boldsymbol{\theta}_j^{(D_{II})}}) = \sum_{j=1}^{M} \mathcal{L}_{\tau_j^{(D_{II})}}\left(f_{\boldsymbol{\theta}' - \alpha \nabla_{\boldsymbol{\theta}'} \mathcal{L}_{\tau_j^{(D_{II})}}}\right). \tag{18}$$

Step ③: We sum up the two domain-specific losses $F(\cdot)$ and $G(\cdot)$, and then update the meta-parameters $\boldsymbol{\theta}$, which emulates the real-time train-test domain shifts and helps the model generalize faster after a few iterations.

These steps are repeated iteratively by randomly sampling from the two different training domains. We provide a performance analysis below to better understand how MAML-DG works. The objective function of MAML-DG is

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= F(\boldsymbol{\theta}) + wG(\boldsymbol{\theta}') \\ &= F(\boldsymbol{\theta}) + wG\left(\boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{\tau_i^{(D_I)}} \mathcal{L}_{\tau_i^{(D_I)}}\left(f_{\boldsymbol{\theta}_i^{(D_I)}}\right)\right) \\ &= F(\boldsymbol{\theta}) + wG\left(\boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})\right),\end{aligned} \tag{19}$$

where $F(\boldsymbol{\theta})$ is the loss function of the first training domain, whereas $G(\boldsymbol{\theta}')$ is that of the second training domain, with $\boldsymbol{\theta}' = \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})$ serving as its initialization.

Applying the first-order Taylor's expansion, we derive that:

$$\begin{aligned}G\left(\boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})\right) &= G(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta}) \cdot (-\beta \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})) \\ &= G(\boldsymbol{\theta}) - \beta \left(\nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})\right).\end{aligned} \tag{20}$$

Note that the remainder of the above Taylor's expansion is:

$$Rem = \frac{1}{2}\left((-\beta \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}))^T \cdot \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} G(\delta) \cdot (-\beta \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}))^T\right), \tag{21}$$

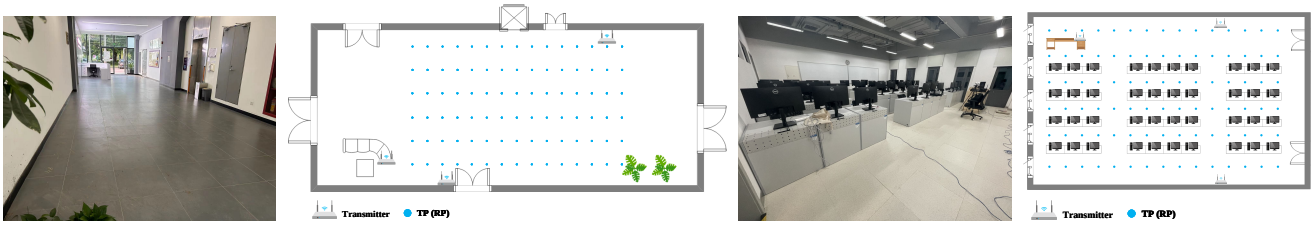
where δ is a number that lies in between $\boldsymbol{\theta}$ and $\boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})$. Plugging Eq. (20) into Eq. (19) yields:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= F(\boldsymbol{\theta}) + wG\left(\boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})\right) \\ &= F(\boldsymbol{\theta}) + wG(\boldsymbol{\theta}) - w\beta \left(\nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})\right).\end{aligned} \tag{22}$$

The loss function is composed of two parts: (i) $F(\boldsymbol{\theta}) + wG(\boldsymbol{\theta})$, and (ii) $-w\beta(\nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}))$. Minimizing this loss function is equivalent to minimizing both (i) and (ii). Part (i) aims to minimize the loss in both training domains, which is intuitive. Part (ii) is equivalent to maximizing the dot product of the gradients of $\nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})$. In other words, we aim to maximize $||\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})||_2 \cdot ||\nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta})||_2 \cdot cos(\delta)$, where δ represents the angle between $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta})$. Therefore, the dot product will be larger if $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}} G(\boldsymbol{\theta})$ tend to have a similar direction. By combining (i) and (ii), the optimizer aims to optimize the losses of both training domains in a similar direction. In contrast, the generally used objective function $F(\boldsymbol{\theta}) + G(\boldsymbol{\theta})$ may cause overfitting by finding a descending root that achieves fast decrease on one domain but slow convergence on the other. MAML-DG can alleviate overfitting and achieve good generalization across different training domains compared to the vanilla MAML.

IV. Experimental Setup

This section describes the data and the baseline models/methods introduced for comparison. Our datasets include



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

Fig. 6. Photographs and layouts of the two scenarios. (a) The photograph of the hall. (b) The layout of the hall. (c) The photograph of the lab. (d) The layout of the lab. In the historical environments, 90 points are considered RPs, while in the new environment, the 90 points are considered TPs.

both synthetic ones generated from computer simulations and the real ones collected in the site surveys on campus at The Chinese University of Hong Kong, Shenzhen.

A. Synthetic Data from Computer Simulations

We generate synthetic data from two different perspectives: the 3rd Generation Partnership Project (3GPP) specifications [53] and the Wireless Insite (WI) platform [54]. The 3GPP specifies propagation models that have been validated by extensive measurement campaigns for typical indoor scenarios. To better reflect real-world propagation conditions, we use the WI platform, which is a suite of ray-tracing models developed by REMCOM that can simulate and predict complex scenarios using advanced electromagnetic processing methods.

From the 3GPP perspective, we classify communication links into line-of-sight (LOS) and non-line-of-sight (NLOS), with probabilities given by Eq. (1) of [55]. We present some toy examples by considering multiple path-loss models that capture signal propagation properties in various indoor environments, including:

- Model (a): Vanilla log-distance model; see Eq. (1) of [45];
- Model (b): Shopping malls with NLOS dual slopes; see Eq. (7) of [55];
- Model (c): Office with mixed LOS and NLOS single slope [53];
- Model (d): Office with a frequency-dependent path-loss exponent; see Eq. (2) and Eq. (5) of [55];
- Model (e): Shopping malls with mixed LOS and NLOS dual slope; see Eq. (2) and Eq. (8) in [55].

We randomly deploy 24 APs operating at 2440 MHz as transmitters. The number of RPs ranges from 10 to 54, and each TP is characterized by its 5-nearest RPs in the signal space. We assume that the data generated from the same path-loss models forms one task.

From the WI perspective, we conduct experiments on six different scenarios generated from the WI platform, in which the layout of the scenarios is designed quite differently. Twenty APs are deployed randomly as transmitters in each scenario. The RSS data generated from scenarios 1, 2, 5 and 6 (S1, S2, S5, S6) are chosen as the training tasks, and the RSS data generated from scenarios 3 (S3) and 4 (S4) are chosen as the test tasks.

B. Real Data from Site Surveys

We conduct site surveys to prototype MetaLoc using a Nexus-5 smartphone as the receiver and three different types of routers as transmitters: the ASUS RT-AC86U, TPlink TL-WR885N, and TPlink TL-WR886N. The entire system operates at 5 GHz with a bandwidth of 20 MHz to ensure high-quality wireless channels.

We perform experiments in two different scenarios depicted in Fig. 6. Specifically, Fig. 6 (a) and (b) present a hall with a test area of 12 m by 5 m, which is almost empty and can be regarded as a pure LOS environment. Inevitably, there will be people walking and elevators opening or closing. We uniformly selected 90 grid points for data collection, where the distance is 0.6 m between any two adjacent points. We performed a CSI measurement campaign from June 9th to July 14th, 2022. Fig. 6 (c) and (d) show a lab room of size 10 m by 8 m, where the test area can be regarded as an NLOS environment, since many obstacles, such as desks and computers, are deployed in the room and they blocked the LOS transmission. We selected 90 grid points for data collection. Unlike the hall, the resolution of the grids of lab is inconsistent. Specifically, the interval between two grid points is 0.6 m or 1.2 m. We collected CSI data on five days from July 21st to September 21st, 2022. Each collection is subject to environmental change.

We collected a total of $(k_{spt}+k_{qry}+1)$ CSI images at each grid point in the new environment during the meta-test stage. We collect only one CSI image for the estimated TP and adopt the histogram intersection metric in Eq. (2) to select its K-nearest RPs. We then form the test task T, which comprises K RPs with a support set containing k_{spt} CSI images and a query set containing k_{qry} CSI images.

C. Baseline Models/Methods

We utilize two different neural network architectures, namely multi-layer perceptron (MLP) and CNN, to implement the proposed MetaLoc framework. We compare MetaLoc with state-of-the-art models/methods, including transfer learning (TL), random initialization (RI), KNN, ILCL, and ConFi, based on real site survey data. Here are the details of each method:

- KNN [15]: We adopt the Euclidean distance metric to select closest $K = 5$ RPs for the estimated TP in the signal space. The averaged locations of the selected RPs are then treated as the estimation result.

TABLE III
LOCALIZATION RESULTS IN THE NEW ENVIRONMENT USING REAL DATA COLLECTED FROM THE SITE SURVEYS

Methods	Hall				Lab			
	Mean errors (m)	Std (m)	Data	Training Steps	Mean errors (m)	Std (m)	Data	Training Steps
MAML (ours)	**2.11**	**1.17**	**3×5**	**7500**	**3.10**	**1.43**	**3×5**	**7500**
MAML-DG (ours)	**2.07**	**1.11**	**3×5**	**2500**	**3.04**	**1.39**	**3×5**	**2500**
MAML-TS (ours)	**2.09**	**1.18**	**3×5**	**5000**	**3.09**	**1.35**	**3×5**	**5000**
TL	2.27	1.27	3×5	7500	3.97	1.99	3×5	7500
RI	2.59	1.29	3×5	7500	4.19	1.95	3×5	7500
ConFi	2.89	0.48	260×90	7500	3.53	0.47	260×90	7500
ILCL	3.61	2.06	3 × 90	7500	3.48	1.62	3 × 90	7500
KNN	2.73	1.35	10 × 90	/	3.35	1.42	10 × 90	/

- TL [56]: Instead of following the standard procedures in the vanilla MAML during the meta-training stage, we adopt the traditional SGD method to train the neural network and obtain the best network parameters for initializing the new environments.
- RI: We randomly generate a set of network parameters for initializing the new environment. The task format and the hyperparameters of the neural network remain the same as those used in the MetaLoc.
- ILCL [36]: We set the number of incremental steps of the BLS classification regression to 10, while the other hyperparameters remain the same as in [36].
- ConFi [32]: We keep the architecture of the neural network and the dataset involved in the training and test procedures the same as those in MetaLoc.

V. EXPERIMENTAL EVALUATIONS

In this section, we formulate localization as a regression problem and present some preliminary results of the toy examples using synthetic data generated from computer simulations. We also formulate localization as a classification problem and verify the efficacy of the proposed MetaLoc based on real site-surveyed data collected from hall and lab shown in Fig. 6.

A. Toy Examples Based on Computer Simulations

For each task in the simulation, we assume the number of support samples k_{spt} is approximately 100, and the number of query samples k_{qry} is 30. As for the training procedure, we set the step size of the inner loop to be $\alpha = 0.0001$ and the step size of the outer loop $\beta = 0.001$. A neural network architecture of MLP, consisting of four hidden layers, is considered to formulate localization as a regression problem, with observed RSS fingerprints shown in Fig. 2 as input and the corresponding locations as output.

1) Convergence Speed: To test the convergence speed of MetaLoc, we exploit a vanilla log-distance path-loss model operating at different standard deviation (std) values to generate synthetic data with the experimental settings given in Table I of [1], which lists four training tasks and one test task with various transmit powers, path-loss exponents, antenna gains and noise levels. In the double-axis system as shown in Fig. 7, the red-axis system represents MetaLoc, while the blue-axis system represents the baseline method RI that was trained using the same neural network architecture but with random initialization of the network parameters. Fig. 7 presents the relationship between RMSE and the number of gradient steps with different training data size. Specifically, the red curve shows that MetaLoc converges much faster than those RI curves. Moreover, Fig. 7 shows that the RI method can be largely affected by the training data size, i.e., the RI performance improves with an increasing scale of training data. When the data size rises to 8000 samples, the localization performance becomes saturated, reaching a level slightly inferior to that of MetaLoc but the latter merely requires only 124 data samples for training. Compared with the traditional fingerprinting methods that strongly rely on the large amount of data collected in the target environment, the MetaLoc framework exploits the existing database built for a batch of different scenarios.

Upper bound of $ER(\boldsymbol{\theta}_T(Q))$ in Theorem I explains the rapid adaptation of MetaLoc from the perspective of parameter space: the smaller distance between $\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}_T^*$, the smaller excess risk, guaranteeing good test performance of $\boldsymbol{\theta}_T(Q)$ on its corresponding task T. We further verify MetaLoc by computing Euclidean distance between $\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}_T^*$ and the resultant distance is 0.48, while for RI cases, the average distance between the randomly initialized parameters and $\boldsymbol{\theta}_T^*$ is 2.12, which show meta-parameters $\boldsymbol{\theta}^*$ locate close to the optimal parameters $\boldsymbol{\theta}_T^*$ and thus facilitate rapid adaptation to new scenarios.

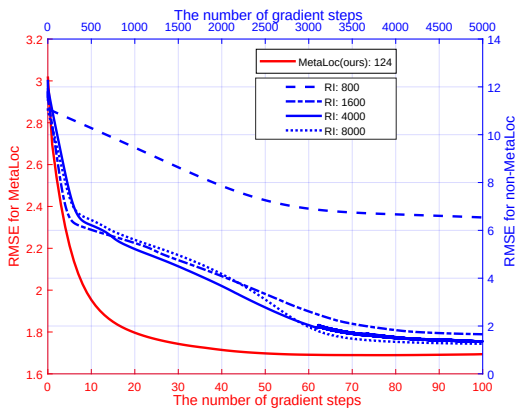


Fig. 7. RMSE convergence comparisons between the MetaLoc and the baseline method RI, which is trained using the same neural network architecture but with the random initialization of the network parameters. In the double-axis system, the red-axis system represents MetaLoc with 124 training data, while the blue-axis system represents RI with 8000, 4000, 1600 and 800 training data.

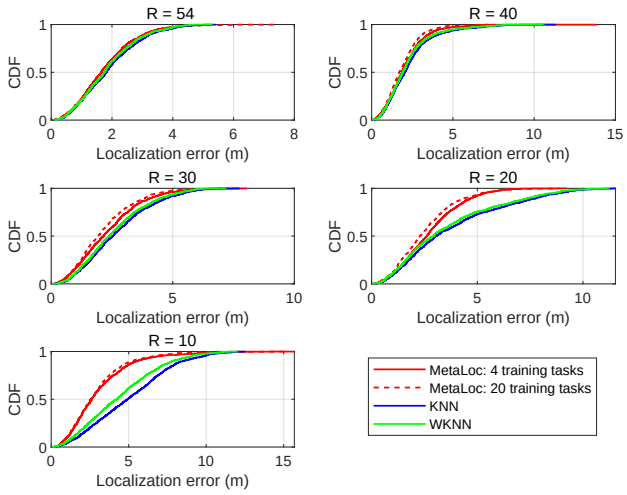


Fig. 8. CDFs of localization errors (m) under different RP numbers (R).

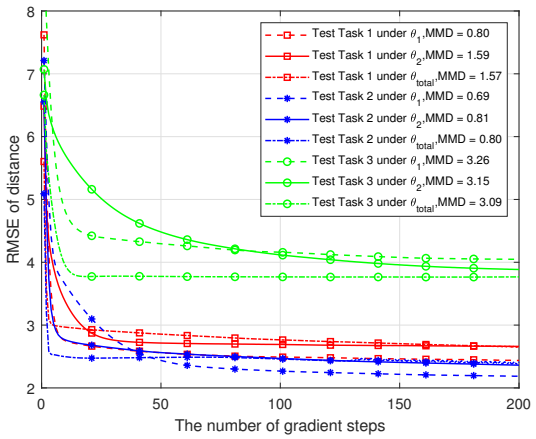


Fig. 9. The test result comparisons among three different meta-parameters, where $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and $\boldsymbol{\theta}_{total}$ represent the well-trained meta-parameters based on two environments, i.e., Environment One, Environment Two, and the total training tasks listed in Table II of reference [1], respectively.

2) Localization Accuracy: We randomly sample 1000 test samples in the test task and quantify the localization errors in terms of the cumulative distribution function (CDF). Fig. 8 presents the CDF of the localization errors versus different RP numbers (R), where $R = 10, 20, 30, 40, 54$. We observe that more RPs deployed in the scenario can promote localization accuracy due to the abundant characteristics of the multipath channel fed into the network. More specifically, when the number of RP $R = 54$, the localization results of the traditional KNN, WKNN and our MetaLoc are nearly the same. As R decreases, the performance gap becomes more significant, and MetaLoc shows higher resistance to performance degradation. When R decreases to 10, the probability of the localization errors of MetaLoc, WKNN and KNN being less than 5 m are 0.89, 0.65 and 0.51 out of one, respectively. MetaLoc presents better accuracy in the severe case of $R = 10$. The above findings indicate that MetaLoc outstandingly reduces the dependence on a large number of RPs and shows the best cost-effectiveness in constructing a fingerprint database.

3) Impact of environment-specific meta-parameters: We utilize multiple path-loss models from 3GPP to characterize different environments instead of the vanilla log-distance model. In Table II of [1], various training tasks with different transmit powers, room sizes, and noise levels are listed. We divide them into three environments based on the path-loss model generation. Environment One is generated from Model (b), Environment Two from Model (c), and Environment Three from Model (d). For the test tasks, we simulate scenarios with a square layout of size 10 m by 10 m and a transmit power of $P_t = 10$ dBm. Test Task One is simulated from Model (e) with $\sigma_{LOS} = 3$, $\sigma_{NLOS} = 6.26$; Test Tasks Two and Three are generated from Model (b) with $\sigma_{NLOS} = 5$ and $\sigma_{NLOS} = 15$, respectively. It should be noted that both Model (b) and Model (e) simulate shopping mall scenarios. MMD measures the average difference between each test task and the training environments, as shown in Fig. 9.

In Fig. 9, $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ represent the environment-specific meta-parameters trained on Environment One and Environment Two, and $\boldsymbol{\theta}_{total}$ are trained on all three environments listed in Table II of [1]. It can be observed that MMD reflects the quality of the test performance, with test tasks with smaller MMD values achieving better localization results. The learned meta-parameters exhibit rapid convergence on multiple test tasks, indicating good generalization ability to new scenarios. Furthermore, in the test tasks with a common noise standard deviation proposed in 3GPP, the environment-specific meta-parameters $\boldsymbol{\theta}_1$ outperform $\boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_{total}$. This is because Environment Two and Environment Three, both generated from office scenarios, cannot provide much specific assistance for test tasks simulated in malls and may even introduce outliers. However, as the noise standard deviation added to the test tasks becomes significantly large, there is no noticeable improvement in the test tasks on $\boldsymbol{\theta}_1$, which has limitations in special cases with extreme noise standard deviation inputs. Overall, these results indicate that MMD can provide preliminary information about the task similarity to assist in selecting environment-specific meta-parameters and to facilitate further improvement in localization accuracy.

4) Wireless Insite (WI): Fig. 10 compares the RMSE results for the data generated from the WI platform under the well-trained parameters obtained by the proposed system. We consider four cases as shown in Fig. 10. Specifically, S3, S4 based on WI&PLM represent the well-trained initialization trained from the path-loss models (abbreviated as PLM in the figure) and WI platform, while S3, S4 based on WI represent the well-trained initialization from the WI platform only. We observe that the results converge extremely fast in the first 50 iterations under all four cases. Interestingly, the solid lines are observed to converge more quickly than the dot lines. A possible explanation for this might be that the well-trained initialization from the path-loss models and WI have more knowledge of the channel features than those trained only over the WI data. Our findings confirm that simulation data can provide extra support in alleviating the data-hungry nature of data-driven localization methods.

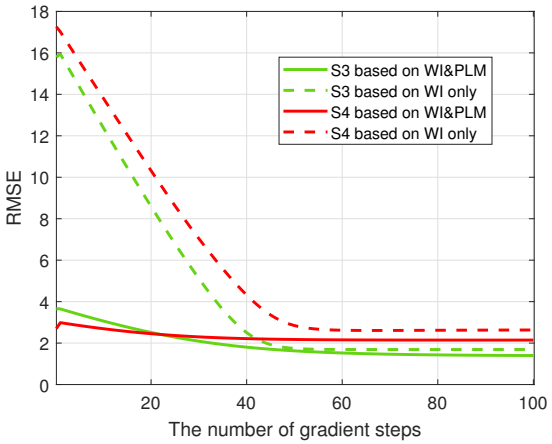
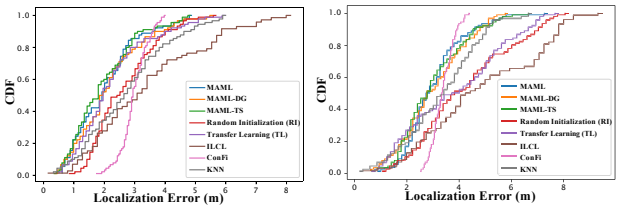


Fig. 10. Comparison results for data generated from the WI platform where PLM represents path-loss models.

B. Comprehensive Results Based on Real Site Surveys

For each task in the site surveys, we set the number of points $N = 10$, the number of support data $k_{spt} = 3$ and the number of query data $k_{qry} = 5$. A CNN, which includes five convolution layers, pooling layers and a fully connected layer, is considered a neural network architecture that formulates localization as a classification problem. The observed CSI fingerprints shown in Fig. 3 are taken as input and the location probabilities as the output. During the meta-training stage, the step size of the inner loop α and the outer loop β are set to 0.01 and 0.001, respectively. In addition, we set the number of gradient descent steps for the inner loop to 5. We collect data on five days, four of which are used for training tasks (i.e., historical environments), and data collected on the fifth day are used for test tasks (i.e., new environments).

1) Localization errors: Table III and Fig. 11 illustrate the localization results of the different methods using real data in the site surveys. We observe that the proposed MAML-DG shows the lowest mean errors of 2.07 m and 3.04 m in the hall and lab, respectively. There are two reasons for the decrease in the localization performance in the lab. First, the propagation environment of the lab is complex with office facilities, and the obstacles exacerbate the multiple path effects. Second, the grid resolution in the lab is inconsistent with that in the hall due to the setup of office cubicles, with a spacing of 1.2 m in the former and 0.6 m in the latter. The coarse resolution comes with a performance penalty in the lab. Performance degradation is particularly evident in the baseline methods, TL and RI, because their initializations were not effectively learned in harsher environments during the training stage. It is noticed that the baseline method ILCL shows improvement in the lab, but the positioning results are somewhat low. The ILCL method is susceptible to the regularization term and data volume. Even when the regularization term is adjusted to the optimal value, the model tends to overfit when the number of CSI images of each test point is too small. Compared to ILCL, MetaLoc is more robust and can quickly adapt to a new environment with only a small sample size. The competing method ConFi presents the smallest variance of



(a) Hall (b) Lab

Fig. 11. CDF comparison of localization errors in the hall and the lab.

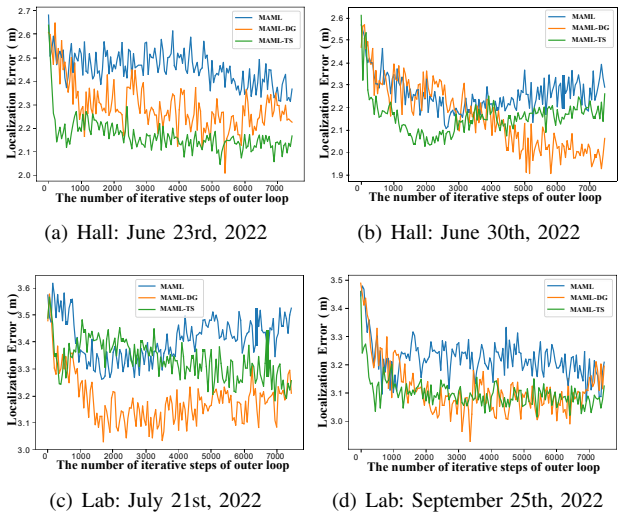
the localization errors, but requires a large amount of data for training in the new environment. Meanwhile, the lowest localization error of ConFi is up to 1.8 m, which is far higher than any other method. The performance of KNN lies in the middle, which is as expected since KNN has poor tolerance to outliers generated due to the fast-changing environments.

In the following, we will delve deeper into the convergence of the proposed framework during both the meta-training and meta-testing stages, illustrated in Fig.12 and Fig.13, respectively. The former demonstrates the impact of the meta-parameters during the meta-training stage, while the latter showcases the efficacy of the well-trained meta-parameters in adapting to a new environment.

2) Convergence of localization errors: During the meta-training stage, the localization errors with respect to the number of iterative steps of the outer loop for the three paradigms of MetaLoc, i.e., vanilla MAML, MAML-DG and MAML-TS, are shown in Fig. 12. Overall, MAML-TS and MAML-DG can achieve faster convergences and smaller localization errors than vanilla MAML. However, MAML-TS sometimes suffers from overfitting since it is only trained in one specific environment. In contrast, as shown in Fig. 12 (b), MAML-DG effectively avoids the overfitting problem because it captures the domain differences from the various environments.

3) Convergence of test errors: During the meta-test stage in a new environment, the well-trained meta-parameters are obtained, and the convergence results of the test errors with respect to the number of gradient steps are shown in Fig. 13. As demonstrated in the figure, the proposed MetaLoc outperforms the baseline methods TL and RI, requiring only a few gradient steps to converge to satisfactory performance. This showcases the extraordinary environmental adaptation abilities of MetaLoc. Moreover, our proposed MAML-DG requires fewer gradient steps to converge than both MAML and MAML-TS.

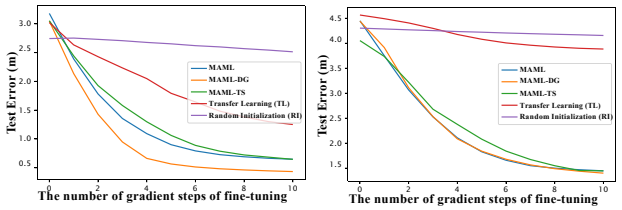
In summary, MetaLoc demonstrates exceptional environmental adaptation abilities, achieving fast adaptation on test tasks with only a small amount of data. In the distributed paradigm, both MAML-DG and MAML-TS algorithms exhibit faster convergence and smaller localization errors than the vanilla MAML in most cases. While the advantage of MAML-DG over MAML is significant, MAML-TS may sometimes suffer from inferior performance due to overfitting issues. Compared to the standard TL method, MetaLoc offers superior performance in terms of smaller localization errors and significantly faster convergence speed, making it a highly promising



(a) Hall: June 23rd, 2022 (b) Hall: June 30th, 2022

(c) Lab: July 21st, 2022 (d) Lab: September 25th, 2022

Fig. 12. Convergence comparison of the localization errors in the hall and the lab.



(a) Hall (b) Lab

Fig. 13. Convergence comparison of test errors in the hall and the lab.

choice for environmental adaptation.

VI. Conclusion

The MetaLoc framework proposed in this paper provides a robust solution for fingerprinting-based localization in dynamic and uncertain environments, ensuring efficient wireless localization. The framework includes two MetaLoc paradigms that leverage past knowledge to fine-tune meta-parameters, resulting in significant improvements in localization accuracy, convergence, and cost-effectiveness. Additionally, MetaLoc is highly versatile and can be applied to any model that uses gradient-based training, making it compatible with a range of classification and regression problems. With its superior performance, MetaLoc has promising potential for large-scale deployment in challenging 5G/6G scenarios such as urban canyons, underground parking, and rapid adaptation across different areas.

References

[1] J. Gao, C. Zhang, Q. Kong, F. Yin, L. Xu, and K. Niu, "Metaloc: Learning to learn indoor RSS fingerprinting localization over multiple scenarios," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 3232–3237.

[2] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2418–2428, 2015.

[3] H. Sun and J. Chen, "Propagation map reconstruction via interpolation assisted matrix completion," *IEEE Trans. Signal Process.*, vol. 70, pp. 6154–6169, 2022.

[4] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "TOA-based robust wireless geolocation and cramér-rao lower bound analysis in harsh LOS/NLOS environments," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2243–2255, 2013.

[5] H. Godrich, A. M. Haimovich, and R. S. Blum, "Target localization accuracy gain in MIMO radar-based systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2783–2803, 2010.

[6] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 3, pp. 194–220, 2017.

[7] R. Wei, B. Li, H. Mo, B. Lu, Y. Long, B. Yang, Q. Dou, Y. Liu, and D. Sun, "Stereo dense scene reconstruction and accurate localization for learning-based navigation of laparoscope in minimally invasive surgery," *IEEE. Trans. Biomed. Eng.*, 2022.

[8] S. Jeong, S. Kuk, and H. Kim, "A smartphone magnetometer-based diagnostic test for automatic contact tracing in infectious disease epidemics," *IEEE Access*, vol. 7, pp. 20 734–20 747, 2019.

[9] B. Zhou, A. Liu, and V. Lau, "Successive localization and beamforming in 5G mmWave MIMO communication systems," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1620–1635, 2019.

[10] J. Gante, G. Falcao, and L. Sousa, "Deep learning architectures for accurate millimeter wave positioning in 5G," *Neural Process. Lett.*, vol. 51, no. 1, pp. 487–514, 2020.

[11] Z. Wang, Z. Liu, Y. Shen, A. Conti, and M. Z. Win, "Location awareness in beyond 5G networks via reconfigurable intelligent surfaces," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 7, pp. 2011–2025, 2022.

[12] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2568–2599, 2019.

[13] S. Ali, W. Saad, N. Rajatheva, K. Chang, D. Steinbach, B. Sliwa, C. Wietfeld, K. Mei, H. Shiri, H.-J. Zepernick *et al.*, "6G white paper on machine learning in wireless communication networks," *arXiv preprint arXiv:2004.13875*, 2020.

[14] Z. Xing, J. Chen, and Y. Tang, "Integrated segmentation and subspace clustering for RSS-based localization under blind calibration," in *Proc. IEEE Global Commun. Conf.*, 2022, pp. 5360–5365.

[15] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE Int. Conf. Comput. Commun.*, vol. 2, 2000, pp. 775–784.

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

[17] C.-H. Hsieh, J.-Y. Chen, and B.-H. Nien, "Deep learning-based indoor localization using received signal strength and channel state information," *IEEE Access*, vol. 7, pp. 33 256–33 267, 2019.

[18] F. Yin and F. Gunnarsson, "Distributed recursive Gaussian processes for RSS map applied to target tracking," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 3, pp. 492–503, 2017.

[19] D. Jin, F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "Bayesian cooperative localization using received signal strength with unknown path loss exponent: Message passing approaches," *IEEE Trans. Signal Process.*, vol. 68, pp. 1120–1135, 2020.

[20] R. S. Sinha, S.-M. Lee, M. Rim, and S.-H. Hwang, "Data augmentation schemes for deep learning in an indoor positioning application," *Electronics*, vol. 8, no. 5, p. 554, 2019.

[21] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2107–2116.

[22] L. Von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy *et al.*, "Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 614–633, 2021.

[23] F. Yin, Y. Zhao, F. Gunnarsson, and F. Gustafsson, "Received-signal-strength threshold optimization using Gaussian processes," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2164–2177, 2017.

[24] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *Proc. IEEE Int. Conf. Indoor Position. Indoor Navig.*, 2014, pp. 261–270.

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

[26] C. Wu, J. Xu, Z. Yang, N. D. Lane, and Z. Yin, "Gain without pain: Accurate WiFi-based localization using fingerprint spatial gradient,"

Proc. ACM Interact. Mobile Wearable Ubiquitous Technol., vol. 1, no. 2, pp. 1–19, 2017.
[27] C. Wu, Z. Yang, C. Xiao, C. Yang, Y. Liu, and M. Liu, "Static power of mobile devices: Self-updating radio maps for wireless indoor localization," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2015, pp. 2497–2505.
[28] S.-H. Fang and C.-H. Wang, "A novel fused positioning feature for handling heterogeneous hardware problem," *IEEE Trans. Commun.*, vol. 63, no. 7, pp. 2713–2723, 2015.
[29] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. M. Ni, "FILA: Fine-grained indoor localization," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 2210–2218.
[30] X. Wang, L. Gao, S. Mao, and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2015, pp. 1666–1671.
[31] X. Wang, X. Wang, and S. Mao, "CiFi: Deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 4673–8999.
[32] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18 066–18 074, 2017.
[33] Z. Gao, Y. Gao, S. Wang, D. Li, and Y. Xu, "CRISLoc: Reconstructable CSI fingerprinting for indoor smartphone localization," *IEEE Internet Things J.*, pp. 3422–3437, 2021.
[34] X. Chen, H. Li, C. Zhou, X. Liu, D. Wu, and G. Dudek, "Fidora: Robust WiFi-based indoor localization via unsupervised domain adaptation," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9872 – 9888, 2022.
[35] H. Li, X. Chen, J. Wang, D. Wu, and X. Liu, "DAFI: WiFi-based device-free indoor localization via domain adaptation," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 5, no. 4, pp. 1–21, 2021.
[36] X. Zhu, W. Qu, X. Zhou, L. Zhao, Z. Ning, and T. Qiu, "Intelligent fingerprint-based localization scheme using CSI images for internet of things," *IEEE Trans. Netw. Sci. Eng.*, 2022.
[37] S. He, W. Lin, and S.-H. G. Chan, "Indoor localization and automatic fingerprint update with altered AP signals," *IEEE Trans. Mobile Comput.*, vol. 16, no. 7, pp. 1897–1910, 2017.
[38] S. He and S.-H. G. Chan, "Sectjunction: Wi-Fi indoor localization based on junction of signal sectors," in *Proc. IEEE Int. Conf. Commun.*, 2014, pp. 2605–2610.
[39] Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang, "ARIEL: Automatic Wi-Fi based room fingerprinting for indoor localization," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 441–450.
[40] R. P. Ghozali and G. P. Kusuma, "Indoor positioning system using regression-based fingerprint method," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 8, pp. 231–239, 2019.
[41] T. M. Deist, A. Patti, Z. Wang, D. Krane, T. Sorenson, and D. Craft, "Simulation-assisted machine learning," *Bioinf.*, vol. 35, no. 20, pp. 4072–4080, 2019.
[42] S. Yang, P. Dessai, M. Verma, and M. Gerla, "FreeLoc: Calibration-free crowdsourced indoor localization," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2013, pp. 2481–2489.
[43] F. Yin, Z. Lin, Q. Kong, Y. Xu, D. Li, S. Theodoridis, and S. R. Cui, "Fedloc: Federated learning framework for data-driven cooperative localization and location data processing," *IEEE Open J. Signal Process.*, vol. 1, pp. 187–215, 2020.
[44] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.
[45] A. Bose and C. H. Foh, "A practical path loss model for indoor WiFi positioning enhancement," in *Proc. IEEE Int. Conf. Inf. Commun. Signal Process.*, 2007, pp. 1–5.
[46] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 53–53, 2011.
[47] M. Schulz, D. Wegemer, and M. Hollick. (2017) Nexmon: The C-based firmware patching framework. [Online]. Available: https://nexmon.org
[48] A. D. Hwang, E. C. Higgins, and M. Pomplun, "A model of top-down attentional control during visual search in complex scenes," *J. Vis.*, vol. 9, no. 5, pp. 25–25, 2009.
[49] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
[50] P. Zhou, Y. Zou, X.-T. Yuan, J. Feng, C. Xiong, and S. Hoi, "Task similarity aware meta learning: Theory-inspired improvement on MAML," in *Proc. Uncertainty Artif. Intell.*, 2021, pp. 23–33.
[51] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, 2012.
[52] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.
[53] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," *3rd Generation Partnership Project (3GPP), Tech. Rep. TR 38.901 V14.1.1 Release 14*, 2017. [Online]. Available: http://www.3gpp.org/DynaReport/38901.htm
[54] P. Mededovic, M. Veletic, and Z. Blagojevic, "Wireless insite software verification via analysis and comparison of simulation and measurement results," in *Proc. Int. Conv. MIPRO*, 2012.
[55] K. Haneda, L. Tian, H. Asplund, J. Li, Y. Wang, D. Steer, C. Li, T. Balercia, S. Lee, Y. Kim *et al.*, "Indoor 5G 3GPP-like channel models for office and shopping mall environments," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2016, pp. 694–699.
[56] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, pp. 1–40, 2016.