MetaLoc: Learning to Learn Wireless Localization

Jun Gao, Graduate Student Member, IEEE, Dongze Wu, Feng Yin, Senior Member, IEEE, Qinglei Kong, Member, IEEE, Lexi Xu, Senior Member, IEEE, and Shuguang Cui, Fellow, IEEE

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 new environments, which results in considerable time and effort being wasted.

To address this challenge, we propose MetaLoc, which is the first fingerprinting-based localization framework that leverages the Model-Agnostic Meta-Learning (MAML). Specifically, built on a deep neural network with strong representation capabilities, MetaLoc is trained on historical data sourced from well-calibrated environments, employing a two-loop optimization mechanism to obtain the meta-parameters. These meta-parameters act as the initialization for quick adaptation in new environments, reducing the need for much human effort. The framework introduces two paradigms for the optimization of meta-parameters: a centralized paradigm that simplifies the process by sharing data from all historical environments, and a distributed paradigm that maintains data privacy by training meta-parameters for each specific environment separately. 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 at: https://github.com/WU-Dongze/MetaLoc.

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 [1], [2], [3], [4]. Although existing global navigation satellite systems (GNSS) provide high outdoor localization accuracy, the demands of emerging applications in diverse areas, such as autonomous driving [5], cooperative 3D scene reconstruction [6], and epidemic tracking [7], require higher levels of accuracy, cost-effectiveness, and robustness. Moreover, with the emergence of mmWave and massive MIMO technologies in 5G and 6G communications, there is a growing demand for precise location information [8], [9], [10]. In light of these growing needs, it is crucial to develop a state-of-the-art localization system that can cover both outdoor and complex indoor environments to achieve a full range of high-precision location-based services [11], [12], [13].

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 typically 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 [14] and Horus [15].

Color versions or more figures in this article are available at https://doi.org/10.1109/JASC.2023.3322766.

Digital Object Identifier 10.1109/JASC.2023.3322766.
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 [16], [17], [18]. Machine learning techniques, being data-hungry, demand large amounts of data samples. Thus, building a database for each target indoor localization becomes time-consuming and labor-intensive. To overcome this challenge, various machine learning techniques have been established and applied to indoor localization, such as data augmentation [19], semi-supervised learning techniques [20], and informed machine learning [21].

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 [14], [22], [23]. 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 deploy a machine learning model in a new environment, one often needs to collect extensive data through site surveys, rebuild the fingerprint database, and then 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 pointed out in a recent 6G white paper [12].

A. Related Works

Indoor localization techniques have been in development 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 [15], 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 [18], involves the use of Bayesian networks for cooperative localization based on RSS. DeepFi [29] 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 [25], its temporal weighted version [36], the cosine similarity [37], the Tanimoto similarity [38], 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: In recent years, 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 [39]. ConFi [31] 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 [16]. 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 [24] uses transfer learning to find the appropriate cross-domain mappings and create a homogeneous feature space that contains discriminative information from different domains. CRISLoc [32] also uses transfer learning to reconstruct a high-dimensional CSI fingerprint database based on outdated fingerprints and a few new measurements. Fidora [33] 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 [35] 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 [25] reduces uncertainty in RSS fingerprints by exploiting spatial gradients among multiple locations. CiFi [30] uses phase differences between antenna pairs instead of raw measurements to improve the stability of CSI fingerprints. DFPS [27] combines raw RSS and the difference between AP pairs to enhance robustness against heterogeneous hardware.

Despite the numerous efforts to identify cost-effective alternatives to traditional site surveys for indoor localization,
with fewer site surveys. A cost-effective indoor fingerprint-based localization method coming these challenges highlight the ongoing need to find a balance of fingerprints over space and time. The difficulties in over-generalizing large-scale fingerprint collections into smaller and local environments through a small number of newly collected measurements. The framework uses the meta-parameters learned from historical environments to initialize a neural network, thereby offering an advancement over the vanilla MAML algorithm employed in our conference paper. Secondly, we enrich our validation through additional real-world datasets, thus offering a more comprehensive evaluation of our framework. Lastly, we incorporate insights pertinent to emergent 6G scenarios, suggesting that the philosophy of learning-to-learn has implications for wireless localization, notably in the next-generation intelligent networks.

Overall, 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 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

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

**B. Contributions**

The paper introduces a pioneering localization framework called MetaLoc, which leverages the power of meta-learning to improve fingerprinting-based indoor localization. It comprises two paradigms: (1) a centralized paradigm based on the vanilla model-agnostic meta-learning (MAML) [43] and (2) a distributed paradigm combining MAML with task similarity (MAML-TS) and MAML with domain generalization (MAML-DG). MetaLoc distinguishes itself from existing methods 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. This work expands upon our conference paper [44] with the following aspects. Firstly, we introduce a new algorithm called MAML-DG, which utilizes a dual-component loss function to improve the learning efficiency, thereby offering an advancement over the vanilla
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 organized 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 computer simulations and real-world data, while future work and challenges towards 6G are presented in Section VI. Finally, the paper is concluded in Section VII. 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]:

\[ P_t - P_r = 10n \log(d) - G + 20 \log \left( \frac{4\pi}{\lambda} \right) + X_\sigma, \]

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 \( \lambda \) is the wavelength of the wireless signal. \( X_\sigma \) is a normal random variable with a standard deviation (std) of \( \sigma \), which ranges from 3 dB to 20 dB.

Although the vanilla path-loss model accounts for the impact of distance and environmental factors on signal...
is represented by a complex number, defined as the ratio of received signal to transmitted signal and extracted from the OFDM receivers can reveal the multipath standards, including 802.11a, 802.11n, and 802.11ac. The CSI is widely used technology in the latest wireless communication environments.

strength, it often falls short in accurately describing complex signal attenuation due to the crude nature of the RSS information, making it susceptible to variability in real-world environments.

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. The Intel 5300 CSI tool [46], a commonly used CSI collection toolkit, requires extensive hardware support and a successful connection to each AP. In contrast, we employ Nexmon [47], a more accessible 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

Figure 2 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 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 $L_i$ ($i = 1, 2, \ldots, V$). In the online stage, fingerprints are collected at TPs to estimate the location using localization algorithms. Next, we will delve into our proposed fingerprint 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 the corresponding RPs, as depicted in Fig. 3. In particular, $RSS_{dis_k}^i$ 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. 1. The CSI images, with dimensions $52 \times 50 \times 50$, 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. 1, 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 [47]:

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

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

Figure 4 presents the challenges and our MetaLoc framework. In transmitter-receiver settings, environmental changes like moving obstacles can significantly alter signal propagation and, consequently, location fingerprints. This may invalidate the existing fingerprint database, necessitating the update for new neural network training. 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.

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 $\theta^*$ learned from the meta-training stage provide a 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 $\tau \sim P(\tau)$, 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 consists of $N$ classes, each with $k$ samples. In the meta-training stage, $M$ training tasks, $\{\tau_i\}_{i=1}^M \sim P(\tau)$, are sampled from the distribution and the corresponding datasets are made available to the model. In the meta-test stage, a new test task $T \sim P(\tau)$ 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) achieves this by learning a set of initial parameters $\theta_{MAML}$ 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 $\theta_{MAML}$ as:

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

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 $\tau_i$, respectively. Then the meta-parameters are updated via stochastic gradient descent (SGD):

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

where $\alpha$ and $\beta$ denote the step size of the inner loop and outer loop, respectively. During the meta-test stage, the meta-parameters $\theta_{MAML}$ are fine-tuned to obtain the parameters $\theta_T$ 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 $\mathcal{L}_T(\theta_{MAML})$ computed based on the support set of the test task, as follows:

$$\theta_T \leftarrow \theta_{MAML} - \alpha \nabla_{\theta} \mathcal{L}_T(\theta_{MAML}),$$

In the subsequent sections, we introduce two paradigms for implementing MetaLoc to learn the meta-parameters: 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 [43]. In this paradigm, data from various historical environments are collectively 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 $\theta^*$. 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 $\theta$, which are represented by the parameterized function $f_\theta$. 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 $\tau_i$ is composed of a loss function $\mathcal{L}_{\tau_i}$, a support set $D_{\tau_i}^s$, and a query set $D_{\tau_i}^q$. 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 ①, we train ① with the support set ②, and derive the task-specific loss ③. We obtain task-specific parameters ③ during the inner loop using a one-step gradient descent update, expressed as

\[ \theta' = \theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^s), \] ④

where the hyper-parameter \( \alpha \) represents the step size of the inner loop. The task-specific parameters \( \theta' \) 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 ②. Further evaluation of their performance on the query set ② 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 \( \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q) \). We define the meta-loss as the sum of all task-specific losses, \( \sum_{i=1}^M \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q) \). This meta-loss is also called the meta-objective. We then update the meta-parameters \( \theta \) by minimizing the meta-loss. In this way, the meta-optimization across all \( M \) tasks is given by

\[ \theta^* = \arg \min \theta \sum_{i=1}^M \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q). \]

It is important to note that during the meta-optimization process, the optimization is performed over the meta-parameters \( \theta \), while the meta-objective is computed using the task-specific parameters \( \theta' \). 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 \( \theta \) are updated as follows:

\[ \theta \leftarrow \theta - \beta \nabla_\theta \sum_{i=1}^M \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q), \]

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

\[ \frac{\partial \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q)}{\partial \theta(m)} = \sum_{n=0}^\infty \frac{\partial \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q)}{\partial \theta_i^{(n)}} \frac{\partial \theta_i^{(n)}}{\partial \theta(m)}, \]

and by merging Eq. (④), we have

\[ \frac{\partial \theta_i^{(n)}}{\partial \theta(m)} = \frac{\partial \theta_i^{(n)}}{\partial \theta(m)} - \alpha \frac{\partial \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q)}{\partial \theta^{(n)}} \frac{\partial \theta^{(n)}}{\partial \theta(m)} \].

However, directly computing the partial derivative 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 \[48\], assuming that the second-order derivatives are negligible. Specifically, we approximate \( \frac{\partial \theta_i^{(n)}}{\partial \theta(m)} \) as follows:

1) When \( m \neq n \):

\[ \frac{\partial \theta_i^{(n)}}{\partial \theta(m)} \approx -\alpha \frac{\partial \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q)}{\partial \theta^{(n)}} \frac{\partial \theta^{(n)}}{\partial \theta(m)} \approx 0. \]

2) When \( m = n \):

\[ \frac{\partial \theta_i^{(n)}}{\partial \theta(m)} \approx 1 - \alpha \frac{\partial \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q)}{\partial \theta^{(n)}} \frac{\partial \theta^{(n)}}{\partial \theta(m)} \approx 1. \]

Using these approximations, Eq. (④) can be simplified to:

\[ \frac{\partial \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q)}{\partial \theta^{(m)}} \approx \frac{\partial \mathcal{L}_{\tau_i}(f_\theta; D_{\tau_i}^q)}{\partial \theta_i^{(n)}} \] ④

Therefore, the meta-optimization over the meta-parameters \( \theta \) can be simplified to optimization over the task-specific parameters \( \theta' \), which eliminates the need for complicated second-order derivatives.

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

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

and task-specific optimal parameters \( \theta_T \) for test task \( T \) is given by

\[ \theta_T = \arg \min \mathcal{L}_T(f_\theta; D_T^q), \]

where \( \mathcal{L}_T(f_\theta; D_T^q) \) denotes the test loss measured on the query set \( D_T^q \). From the process outlined above, it is clear that this procedure exhibits a model-agnostic characteristic. Specifically, it does not depend on a particular structure of the model. Instead, the process is solely dictated by the model’s parameters and their gradients, indicating its potential use with any model that utilizes a gradient descent type optimization method for training. For clarity, the entire process is summarized in Algorithm 1.

3) Performance Analysis of the Vanilla MAML: While Finn et al. [43] demonstrated the effectiveness of the vanilla MAML in various tasks, they did not provide a comprehensive theoretical analysis to support its effectiveness using few-gradient based adaptation. Later, Zhou et al. provided performance guarantees and some deeper insights into the superiority of MAML over traditional neural network training [49]. We borrow similar ideas of reference [49] to address the following questions: (1) what is the reason behind the superior performance of the vanilla MAML over traditionally...
Algorithm 1 Vanilla MAML

Require:
\( \mathcal{P}(\tau) \): distribution over tasks;
\( \alpha \): step size of the inner loop;
\( \beta \): step size of the outer loop;

Meta-training Stage (in the historical environments):
1: Randomly initialize \( \theta \);
2: For \( i \in \mathcal{I} \) in iterations do:
3: \( \tau \sim \mathcal{P}(\tau) \);
4: For each \( i \in \{1,2,\ldots,M\} \) do:
5: \( \theta_{i} = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\tau}(f_{\theta}; D_{\tau}^{s}) \);
6: \( \theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\tau} \mathcal{L}_{\tau}(f_{\theta}; D_{\tau}^{s}) \);
7: return \( \theta^{*} \leftarrow \theta \) when it converges.

Meta-test Stage (in the new environment):
8: Sample a test task \( T \sim \mathcal{P}(\tau) \);
9: \( \theta_{T} \leftarrow \theta_{T} - \alpha \nabla_{\theta} \mathcal{L}_{T}(f_{\theta}; D_{T}^{s}) \);
10: return \( \theta_{T}^{*} \leftarrow \theta_{T} \) when it converges.

Following the above idea, we compare the vanilla MAML with a conventional neural network in the next. The vanilla MAML trains the meta-parameters \( \theta \) 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 \( \theta_{T}^{i}, i = 1,\ldots,M \). 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 determine the direction towards the optimal parameters for each training task (represented in different colors). The meta-parameters \( \theta \) are then updated based on the average across the optimal directions of these training tasks (path shown in black). Through this approach, MAML updates the meta-parameters \( \theta \) in a direction that aligns with all training tasks, with each \( \theta_{T}^{i} \), receiving equal weight in the gradient descents of the outer loop. Consequently, it is expected that

\[
E_{T \sim \mathcal{P}} \| \theta^{*} - \theta_{T}^{*} \|_{2}^{2} \]

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, \( E_{T \sim \mathcal{P}} \| \theta^{*} - \theta_{T}^{*} \|_{2}^{2} \) would be larger than in MAML, as demonstrated in Section V-A.

C. Distributed MetaLoc Paradigm

Compared to the centralized paradigm, which is trained on all historical environments, our conference work [44] 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 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)
Algorithm 2 MAML-TS

Require:
\{\mathcal{P}^{(i)}(\tau)\}_{i=1}^{S}\}: distributions over tasks in \(S\) domains;
\(\alpha\): step size of the inner loop;
\(\beta\): step size of the outer loop;

Meta-training Stage (in the historical environments):
1: Randomly initialize \(\theta\);
2: For environment \(i\) in \(\{1, 2, \ldots, S\}\) do:
3: For \(ite\) in iterations do:
4: Sample training tasks \(\{\tau_i\}_{j=1}^{M_i} \sim \mathcal{P}(\tau)\);
5: For each \(j\) in \(\{1, 2, \ldots, M\}\) do:
6: \(\theta'_j = \theta - \alpha \nabla_{\theta} L_{\tau_j}(f_{\theta}; D_{\tau_j})\);
7: \(\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\tau_j} L_{\tau_j}(f_{\theta_j}; D_{\tau_j});\)
8: \(\text{return } \theta^* \) when it converges.

Meta-test Stage (in the new environment):
9: Sample a test task \(T \sim \mathcal{P}(\tau)\);
10: Compute \(\text{MMD}^{[50]}\) between the test task and training tasks from each historical environment \(\{\tau_i\}_{j=1}^{M_i} \sim \mathcal{P}(\tau), i = 1, 2, \ldots, S\);
11: Choose the optimal environment-specific meta-parameters \(\theta^*_i\), where \(i^* = \arg\min_{i \in \{1, \ldots, S\}} \text{MMD}\).
12: \(\theta_T \leftarrow \theta^*_i - \alpha \nabla_{\theta} L_T(f_{\theta}; D_{\tau_i});\)
13: \(\text{return } \theta_T^* \) when it converges.

Algorithm 3 MAML-DG

Require:
\{\mathcal{P}^{(i)}(\tau)\}_{i=1}^{S}\}: distributions over tasks in \(S\) domains;
\(\alpha\): step size of the inner loop;
\(\beta\): step size of the outer loop;
\(w\): weight of the loss function of the second training domain \(D_{11}\);

Meta-training Stage (in the historical environments):
1: Randomly initialize \(\theta\);
2: For \(ite\) in iterations do:
3: Sample two training domains \(D_I\) and \(D_{11}\) uniformly from \(\{1, 2, \ldots, S\}\);
4: Sample tasks \(\{\tau_i^{(D_I)}\}_{j=1}^{M_i} \sim \mathcal{P}(\tau)\) in domain \(D_I\);
5: For \(i\) in range \(M\) do:
6: \(\theta_i^{(D_I)} = \theta - \alpha \nabla_{\theta} L_{\tau_i^{(D_I)}}(f_{\theta}; D_{\tau_i^{(D_I)}})\);
7: \(\theta^* = \theta - \beta \nabla_{\theta} \sum_{\tau_j^{(D_I)}} L_{\tau_j^{(D_I)}}(f_{\theta_i^{(D_I)}}; D_{\tau_j^{(D_I)}})\);
8: Sample tasks \(\{\tau_j^{(D_{11})}\}_{i=1}^{M} \sim \mathcal{P}(\tau)\) in \(D_{11}\);
9: For \(j\) in range \(M\) do:
10: \(\theta_j^{(D_{11})} = \theta - \alpha \nabla_{\theta} L_{\tau_j^{(D_{11})}}(f_{\theta_j^{(D_{11})}}; D_{\tau_j^{(D_{11})}})\);
11: \(\theta \leftarrow \theta^* - w \beta \nabla_{\theta} \sum_{\tau_j^{(D_{11})}} L_{\tau_j^{(D_{11})}}(f_{\theta_j^{(D_{11})}}; D_{\tau_j^{(D_{11})}})\);
12: \(\text{return } \theta^* \) when it converges.

Meta-test Stage (in the new environment):
13: Sample a test task \(T \sim \mathcal{P}(\tau)\);
14: \(\theta_T \leftarrow \theta^* - \alpha \nabla_{\theta} L_T(f_{\theta}; D_{\tau_i})\);
15: \(\text{return } \theta_T^* \) when it converges.

---

where \(x\) and \(y\) are wireless signal features, such as RSS fingerprints or CSI fingerprints, for a pair of localization tasks \(\tau_1\) and \(\tau_2\) in the tasks distribution \(\mathcal{P}(\tau)\), respectively. Moreover, \(\mathcal{H}\) refers to a class of functions \(h\). In [50], 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 \(\text{MMD}[\mathcal{H}, \tau_1, \tau_2] = 0\) if and only if \(\tau_1\) is equivalent to \(\tau_2\). A higher \(\text{MMD}\) value indicates a larger difference between the two tasks. Given that the algorithm takes into account task similarity, we term it as the model-agnostic meta-learning with task similarity (MAML-TS). The algorithm outlined in Algorithm 2, allows training in isolation for each environment, thereby ensuring better data security.

To further improve the learning efficiency, 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 [51] 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.

As outlined in Algorithm 3, MAML-DG is designed to train a deep learning model with parameters \(\theta\) 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 \(D_I, D_{11} = 1, 2, \ldots, S\) and \(D_{11}, D_{11} = 1, 2, \ldots, S, D_I \neq D_{11}\) and generates tasks in these two domains. The full steps are as follows.

**Step ①**: We virtually train a domain-specific meta-parameters \(\theta^*\) on the tasks generated from the training domain \(D_I\) using the vanilla MAML algorithm. We derive the first domain-specific loss function as

\[
F(\cdot) = \sum_{\tau_i^{(D_I)}} L_{\tau_i^{(D_I)}}(f_{\theta_i^{(D_I)}}) = \sum_{i=1}^{M} L_{\tau_i^{(D_I)}}(f_{\theta_i^{(D_I)}}) - \alpha \nabla_{\theta} L_{\tau_i^{(D_I)}} \bigg|_{\theta_i^{(D_I)}}. 
\]

**Step ②**: With the initialization \(\theta^*\) obtained in the previous step, we derive a second domain-specific loss function for the tasks generated from the training domain \(D_{11}\) using vanilla MAML once again, which is shown as

\[
G(\cdot) = \sum_{\tau_j^{(D_{11})}} L_{\tau_j^{(D_{11})}}(f_{\theta_j^{(D_{11})}}) = \sum_{j=1}^{M} L_{\tau_j^{(D_{11})}}(f_{\theta_j^{(D_{11})}}) - \alpha \nabla_{\theta} L_{\tau_j^{(D_{11})}} \bigg|_{\theta_j^{(D_{11})}}. 
\]

**Step ③**: We sum up the two domain-specific losses \(F(\cdot)\) and \(G(\cdot)\), and then update the meta-parameters \(\theta\), which emulates

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
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

\[ \mathcal{L}(\theta) = F(\theta) + wG(\theta') \]

\[ = F(\theta) + wG(\theta - \beta \nabla F(\theta)) \]

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

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

\[ G(\theta - \beta \nabla F(\theta)) = G(\theta) + \nabla G(\theta) \cdot (\beta \nabla F(\theta)) \]

\[ = G(\theta) - \beta (\nabla G(\theta) \cdot \nabla F(\theta)) \quad (20) \]

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

\[ \text{Rem} = \frac{1}{2} (\beta \nabla F(\theta))^T \nabla G(\theta) \cdot (\beta \nabla F(\theta))^T \]

\[ = \frac{1}{2} \beta^2 \nabla^2 F(\theta)^T \nabla G(\theta) \cdot \nabla F(\theta) \quad (21) \]

The loss function is composed of two parts: (i) \( F(\theta) + wG(\theta') \), and (ii) \( -w\beta \nabla G(\theta) \cdot \nabla F(\theta) \). 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 \( \nabla G(\theta) \) and \( \nabla F(\theta) \). In other words, we aim to maximize \( ||\nabla G(\theta)||^2 \cdot ||\nabla F(\theta)||^2 \cdot \cos(\delta) \), where \( \delta \) represents the angle between \( \nabla G(\theta) \) and \( \nabla F(\theta) \). Therefore, the dot product will be larger if \( \nabla G(\theta) \) and \( \nabla F(\theta) \) tend to have a similar direction. In combining aspects (i) and (ii), the optimizer is designed to guide the loss reduction in both training domains towards a similar direction. This strategy contributes to the accelerated convergence of MAML-DG. In comparison, the commonly employed objective function \( F(\theta) + G(\theta) \) could lead to slower convergence, as it seeks a path that allows for rapid decrease in one domain but slower convergence in the other.

D. Comparison Between Paradigms

Figure 6 summarizes our proposed paradigms. For clarity, we utilize three distinct environments as examples, each denoted by differently colored circles. Vanilla MAML depicted in Fig. 6 (a) relies on a comprehensive data collection process from all environments to derive the meta-parameters \( \theta' \). This process necessitates transferring raw wireless signal data, including RSS and CSI, to a remote server, thereby introducing potential privacy concerns. In contrast, the proposed

MAML-TS, displayed in Fig. 6 (b), calculates environment-specific meta-parameters independently within each local environment. In this method, each historical environment has its own local server. The task-similarity MMD between the test dataset and each historical environment is computed in their own local server, ensuring raw data from the historical environments is unexposed, and data privacy is thus maintained. As for MAML-DG shown in Fig. 6 (c), for each iteration we start the process by randomly sampling two representatives from the pool of all environments. The meta-parameters shared across domains, as indicated in lines 7 and 11 of Algorithm 3, are tied to the neural network model and do not explicitly reveal any user-specific data. Although there are established measures to confront such privacy concerns, a detailed investigation of these strategies is outside the scope of our current research [52].

IV. EXPERIMENTAL SETUP

This section describes the data and the baseline models/methods introduced for comparison. Our datasets include 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 REMOCM 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];
A. Toy Examples Based on Computer Simulations

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. 7.

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, TP-Link TL-WR885N, and TP-Link 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. 7. Specifically, Fig. 7 (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. Figure 7 (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.

Training a machine learning model for localization requires a large amount of data. In our simulations, we use site-surveyed data collected from hall and lab shown in Fig. 7. Specifically, Fig. 7 (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. Figure 7 (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 \((k_{\text{SPT}} + k_{\text{QRY}} + 1)\) 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 \(k_{\text{SPT}}\) CSI images and a query set containing \(k_{\text{QRY}}\) 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. Our choice of these specific neural networks stems from their widespread use in the state-of-the-art works. For example, ConFi [31] and CiFi [30] utilized CNNs, while a simple MLP was adopted in [39]. We compare MetaLoc with the state-of-the-art models/methods as follows:

- KNN [14]: 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.
- 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 [35]: We set the number of incremental steps of the BLS classification regression to 10, while the other hyperparameters remain the same as in [35].
- ConFi [31]: We keep the neural network architecture and dataset the same as MetaLoc. Data from historical environments serve as the training set, while data collected in new environment serve as the test set.

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. 7.

A. Toy Examples Based on Computer Simulations

For each task in the simulation, we assume the number of support samples \(k_{\text{SPT}}\) is approximately 100, and the number of query samples \(k_{\text{QRY}}\) is 30. As for the training procedure,
TABLE III
LOCALIZATION RESULTS IN THE NEW ENVIRONMENT USING REAL DATA COLLECTED FROM THE SITE SURVEYS

| Methods     | 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 | /             |

Fig. 8. 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.

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. 3 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 std values to generate synthetic data with the experimental settings given in Table I of [44], 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. 8, the red-axis system represents MetaLoc, while the blue-axis system represents RI with 8000, 4000, 1600 and 800 training data.

![Fig. 8](image)

Fig. 8. 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.

MetaLoc framework exploits the existing database built for a batch of different scenarios.

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

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). Figure 9 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, respectively. MetaLoc presents better accuracy in the severe case of $R = 10$. The above findings are validated in Fig. 9.

![Fig. 9](image)

Fig. 9. CDFs of localization errors (m) under different RP numbers ($R$).
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 [44], 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. 10.

In Fig. 10, $\theta_1$ and $\theta_2$ represent the environment-specific meta-parameters trained on Environment One and Environment Two, and $\theta_{total}$ are trained on all three environments listed in Table II of [44]. 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 $\theta_1$ outperform $\theta_2$ and $\theta_{total}$. 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 $\theta_1$, 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): Figure 11 compares the RMSE results for the data generated from the WI platform under the well-trained parameters obtained by the proposed framework. We consider four cases as shown in Fig. 11. Specifically, S3, S4 based on WI&PLM represent the well-trained initialization 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.

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 $k_{sppt} = 3$ and the number of query data $k_{qry} = 5$. A CNN including 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. 1 are taken as input and the location probabilities as the output. During the meta-training stage, the step size of the inner loop $\alpha$ and the outer loop $\beta$ 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. 12 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 method RI, because the initialization was not effectively learned in harsher environments during the training stage. Moreover, in less complex environment such as the hall scene, the performance gap between TL and our MetaLoc is minor. But in a more complex lab environment, the difference becomes noticeable. This is because environmental complexity magnifies the conflicts among the learned knowledge, impairing the performance of TL. In contrast, the inherent learning-to-learn capability of meta-learning effectively mitigates these knowledge conflicts, thereby maintaining performance even in more complex scenes.

Besides, it is noticed that the baseline method ILCL shows improvement in the lab, but the positioning error are somewhat high. 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 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-test stages, illustrated in Fig. 13 and Fig. 14, 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. 13. 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. 13 (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. 14. 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.

C. Further Discussions About MetaLoc and TL

MetaLoc and TL are both techniques that apply previous learning experiences to new tasks. TL primarily imparts existing knowledge, while MetaLoc conveys learning strategies across diverse tasks. TL can encounter the problem of negative transfer when there is conflict in the shared knowledge. In dynamic environments for wireless localization, where interference, noise, and physical obstructions continuously change the channel conditions, the learning-to-learn ability of MetaLoc, becomes particularly useful. On the contrary, TL strongly depends on the knowledge specific to training environments, which may not work effectively to new scenarios, especially when the condition is very complex, as shown in Fig. 12 (b).
VI. FUTURE WORK AND CHALLENGES TOWARDS 6G

The improvements of MetaLoc in accuracy, robustness, and cost-effectiveness foster the development of further research, particularly in the forthcoming 6G era. Considering large scale and diverse scenarios in 6G, the learning-to-learn capability of MetaLoc becomes a spotlight in this emerging research frontier, paving the way for intelligent, adaptive networks. In this section, we will shed some light on a few representative scenarios and associated challenges.

A. Future Work

1) Joint Communication and Sensing (JCAS): In 6G communication systems, the incorporation of higher frequency bands, wider bandwidths, and massive antenna arrays not only enhances traditional communication services, but also supports high-precision, high-resolution sensing [57], etc. MetaLoc, utilizing the wireless network as a sensor, is capable of rapidly adapting to new environments for precise user localization. This location information, in turn, can improve communication performance in our future work, placing MetaLoc as a cost-effective example of JCAS, especially for dynamic environments. In the future, we plan to extend this framework to fuse wireless signal data that carry information about human postures and the location information, making MetaLoc suitable for applications like smart homes and remote surgeries.

2) Digital Twins: Digital twins offer virtual representations of physical systems through ML, data analytics, and multiphysics simulation [58]. As 6G technology evolves towards self-sustainability and proactive online learning, it aims to minimize human intervention and thus reduce costs. With its inherent capacity of learning-to-learn, MetaLoc is well-suited to this context. In the framework of digital twins, when deviations arise between the simulated and actual environments, model adjustments are necessary. Utilizing accumulated historical data, MetaLoc can swiftly facilitate model recalibration in a cost-effective way. Our research validates that meta-parameters derived from 3GPP-defined environments effectively facilitate learning in WI-defined scenarios, thus reinforcing the capabilities of digital twins. This interplay between MetaLoc and digital twins illuminates the potential of such a combination in advancing 6G self-sustaining wireless systems.

B. Challenges

The procurement of high-quality and substantial training data in the dynamic landscape of 6G networks presents considerable challenges. Each environment, with its unique attributes and inherent interference, may result in data of different forms. The mixture of this heterogeneous data into a coherent system necessitates the utilization of advanced methodologies such as multimodal data fusion techniques [59]. Moreover, before using the data in these algorithms, thorough preprocessing is typically a necessary procedure. This includes steps such as noise filtering, outlier removal, and data normalization, which collectively ensure the preservation of data integrity and the enhancement of its utility [32]. To distill valuable features from the raw data, principal component analysis (PCA) [60] is often utilized. Beyond the realm of traditional techniques, leveraging advanced feature construction methods driven by deep learning [61], [62] could provide a more sophisticated level of feature representation. Moreover, data privacy regulations may potentially restrict the volume and type of data accessible for MetaLoc, and some potential solutions might involve synthetic data generation [63], federated learning [42], each solution requiring further exploration when combined with MetaLoc.

VII. 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 learn 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 as we mentioned in Section VI. We believe that the philosophy of learning-to-learn will extend far beyond the scope of wireless localization, holding substantial relevance for the future large-scale intelligent systems.

REFERENCES

[1] R. Faragher and R. Harle, “Location fingerprinting with Bluetooth low energy beacons,” IEEE J. Sel. Areas Commun., vol. 33, no. 11, pp. 2418–2428, Nov. 2015.
[2] H. Sun and J. Chen, “Propagation map reconstruction via interpolation assisted matrix completion,” IEEE Trans. Signal Process., vol. 70, pp. 6154–6169, 2022.
[3] 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, May 2013.
[4] 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, Jun. 2010.
[5] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” IEEE Trans. Intell. Vehicles, vol. 2, no. 3, pp. 194–220, Sep. 2017.
[6] R. Wei et al., “Stereo dense scene reconstruction and accurate localization for learning-based navigation of laparoscope in minimally invasive surgery,” IEEE Trans. Biomed. Eng., vol. 70, no. 2, pp. 488–500, Feb. 2023.
[7] S. Jeong, S. Suk, and H. Kim, “A smartphone magnetometer-based diagnostic test for automatic contact tracing in infectious disease epidemics,” IEEE Access, vol. 7, pp. 20734–20747, 2019.
[8] 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, Mar. 2019.
[9] J. Gante, G. Falcão, and L. Sousa, “Deep learning architectures for accurate millimeter wave positioning in 5G,” Neural Process. Lett., vol. 51, no. 1, pp. 487–514, Feb. 2020.
[10] 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, Jul. 2022.
[11] F. Zafari, A. Gkelias, and K. K. Leung, “A survey of indoor localization systems and technologies,” IEEE Commun. Surveys Tuts., vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.
[12] S. Ali et al., “6G white paper on machine learning in wireless communication networks,” 2020, arXiv:2004.13875.
Jun Gao (Graduate Student Member, IEEE) received the B.Eng. degree in communication engineering from Nanchang University, Nanchang, China, in 2016, and the M.Eng. degree in electronic and communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2019. She is currently pursuing the joint Ph.D. degree with the Future Network of Intelligence Institute (FNii) and the School of Science and Engineering (SSE), The Chinese University of Hong Kong, Shenzhen, China. Her research interests focus on machine learning, wireless sensing and localization.

Dongze Wu received the B.Sc. degree in mathematics and applied mathematics from The Chinese University of Hong Kong, Shenzhen, China, in 2023. He is currently pursuing the M.Sc. degree in statistical science with the University of Oxford. His research interests include wireless communications, statistical signal processing, and neuroscience.

Feng Yin (Senior Member, IEEE) received the B.Sc. degree from Shanghai Jiao Tong University, China, and the M.Sc. and Ph.D. degrees from Technische Universitaet Darmstadt, Germany. From 2014 to 2016, he was a Marie-Curie Post-Doctoral Researcher with Ericsson Research, Linkoping, Sweden. Since 2016, he has been with The Chinese University of Hong Kong, Shenzhen, as an Assistant Professor. He has published around 90 top-tier journal articles and flag-ship conference papers, more than 20 U.S. and Chinese Patents, with over 1000 citations and more than 30 high-impact publications, and his research interests include statistical signal processing, Bayesian learning and optimization, and sensory data fusion. He was a recipient of the Chinese Government Award for Outstanding Self-Financed Students Abroad in 2013 and the Marie Curie Young Scholarship from the European Union in 2014. He was the Finalist of the IEEE CAMSAP Conference Best Paper Award in 2013 and received the Best Paper Award of Springer ICSINC Conference in 2022. He currently serves as an Associate Editor for the Elsevier Signal Processing Journal and IEEE Transactions on Signal Processing.

Qinglei Kong (Member, IEEE) received the B.Eng. degree in communication engineering from the Harbin Institute of Technology, Harbin, China, in 2012, the M.Eng. degree in electronic and information engineering from the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China, in 2015, and the Ph.D. degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, in 2018. She is currently an Assistant Professor with the Institute of Space Science and Applied Technology, Harbin Institute of Technology (Shenzhen). She used to work with Cyber Security Cluster, Institute for Infocomm Research, Singapore, and Tencent Security, Shenzhen, as a Research Scientist. She was a Post-Doctoral Researcher with The Chinese University of Hong Kong, Shenzhen. Her research interests include applied cryptography, blockchain, VANET, and game theory.

Lexi Xu (Senior Member, IEEE) received the M.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2009, and the Ph.D. degree from the Queen Mary University of London, London, U.K., in 2013. From 2013 to 2020, he was a Senior Engineer with the Network Technology Research Institute, China United Network Communications Corporation (China Unicom). Since August 2020, he has been a Senior Engineer with the Research Institute of China Unicom. He is also a China Unicom delegate in ITU, ETSI, 3GPP, and CCSA. He is also a Professor (part-time) with the Beijing University of Posts and Telecommunications. His research interests include big data, self-organizing networks, satellite systems, and radio resource management in wireless systems.

Shuguang Cui (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Stanford University, CA, USA, in 2005. Afterwards, he has been working as an Assistant Professor, an Associate Professor, a Full Professor, and a Chair Professor in electrical and computer engineering with the University of Arizona, Texas A&M University, UC Davis, and CUHK at Shenzhen, respectively. He has served as the Executive Dean of the School of Science and Engineering, CUHK at Shenzhen, the Executive Vice Director of the Shenzhen Research Institute of Big Data, and the Director of the Future Network of Intelligence Institute (FNii). His current research interests focus on the merging between AI and communication networks. He was selected as the Thomson Reuters Highly Cited Researcher and listed in the World’s Most Influential Scientific Minds by ScienceWatch in 2014. He was a recipient of the IEEE Signal Processing Society 2012 Best Paper Award. He has served as the general co-chair and the TPC co-chairs for many IEEE conferences. He has also been serving as an Area Editor for IEEE Signal Processing Magazine and an Associate Editor for IEEE TRANSACTIONS ON BIG DATA, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is also a member of the Steering Committee of IEEE TRANSACTIONS ON BIG DATA and the Chair of the Steering Committee of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is also a member of the IEEE WiSec Technical Committee and a member of the IEEE WiSec Technical Committee. He is also a member of the IEEE WiSec Technical Committee. He is also a member of the IEEE WiSec Technical Committee. He is also a member of the IEEE WiSec Technical Committee. He is also a member of the IEEE WiSec Technical Committee.