Meta-GLoc: GNN for Adaptive and Robust WiFi Localization with Meta-Learning

Qiqi Xiao, Ziqi Ye, Yinghui He, Jianwei Liu, and Guanding Yu, *Senior Member, IEEE*
College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China
E-mail: {xiaoqiqi, yeziqi, 2014hyh, jianweiliu, yuguanding}@zju.edu.cn

***Abstract*—Recent deep learning-based localization methods leverage meta-learning to enhance adaptability across diverse environments. However, most existing approaches focus on variations in environmental layouts while overlooking the changes in device configurations—such as the number of access points, antennas, or bandwidth. Unlike environmental variations, changes in device configurations fundamentally alter the dimensionality of channel state information (CSI), which can significantly hinder the usability and generalizability of neural networks. To address this problem, we propose Meta-GLoc, a novel WiFi localization system that combines meta-learning with graph neural networks to enhance both localization generalization and robustness. Specifically, we introduce an amplitude-phase fusion method and a feature extraction method to construct fine-grained CSI graphs. The former uses both amplitude and phase to construct CSI images, improving data reliability, while the latter extracts dimension-consistent features to mitigate the impact of varying bandwidth and antenna configurations. Moreover, meta-learning is employed to realize adaptive localization in different environments or configurations. Experiment results on commodity WiFi devices across different configurations demonstrate that Meta-GLoc effectively improves localization accuracy and robustness.**

***Index Terms*—WiFi localization, graph neural network, meta-learning, adaptive localization.**

I. Introduction

Various emerging intelligent applications, such as augmented reality and industrial mobile robots, heavily rely on indoor localization technologies [1]. However, global positioning system is difficult to achieve precise localization indoors due to signal obstruction caused by buildings. To address this issue, WiFi-based localization has attracted significant attention, owing to its widespread availability and large-scale deployment in indoor environments [2]. The channel state information (CSI) extracted from WiFi packets captures the characteristics of the propagation path between the transmitter and receiver, and therefore inherently contains location-related information. To effectively extract this information, especially in non-line-of-sight (NLoS) scenarios, deep learning (DL)-based approaches have been proposed [3], [4]. By leveraging the powerful representation capabilities of neural networks, these methods have achieved high-accuracy localization.

However, the success of DL-based approaches heavily relies on extensive high-quality CSI [5]. A model trained on existing scenarios often generalizes poorly to a new and unseen scenario due to the lack of exposure to data from the new scenario. A straightforward solution is to re-collect data and re-train the model, causing high overheads. To mitigate this issue, meta-learning has been adopted since it develops a generalized initialization that facilitates rapid adaptation to unseen scenarios by training the model on a variety of scenarios [6]–[8]. Nevertheless, prior works typically focus on deployment variation across different environments, e.g., the locations of the access points (APs) and surrounding reflectors [9]–[11], while overlooking the impact of variations in device configurations. In fact, the configurations of the transceiver devices (e.g., the number and bandwidth of the WiFi APs) generally are different across different scenarios, and even dynamically vary in the same scenario for communication purposes. Unlike environmental changes, the variation in the configuration would result in different WiFi CSI dimensions and contained information. Such variation is challenging to address with existing solutions, as it necessitates adjustments to the model structure, which conflicts with the fixed input dimensions required by these methods. This limitation significantly hampers the practical deployment of WiFi localization systems.

To address this issue, we propose a novel WiFi localization system, namely Meta-GLoc. It leverages the scalability and topology-invariance properties of graph neural networks (GNNs) [12] to effectively handle variations in the number of WiFi APs, while incorporating the fast adaptation of meta-learning to achieve efficient cross-scenario localization. By collecting a few CSI samples in different scenarios and utilizing the CSI corresponding to each AP as the input of the graph node to train and fine-tune the GNN, Meta-GLoc can achieve adaptive cross-scenario localization at low cost. However, there are still two challenges. First, the noise and error in the amplitude and phase of the CSI differ in origin and degree. Consequently, feeding the raw CSI directly into the network may degrade its performance. Second, each AP's bandwidth may fluctuate dynamically due to interference from surrounding devices. As a result, the size of the CSI for each AP may still change, making it difficult to be directly input into the same GNN. To tackle these challenges, we introduce an amplitude-phase fusion method and a feature extraction method to construct fine-grained CSI graphs. The former first mitigates noise and errors in the amplitude and phase separately. Then, it constructs a CSI image by combining them in a 2:1 ratio, considering their different reliability, thereby enhancing the localization robustness. The latter extracts bandwidth-independent and dimension-consistent features with spatial pyramid pooling [13] and uses them as graph node inputs for the GNN, thereby resolving the dimensionality issue. The real-world experiment results with a prototype of Meta-GLoc

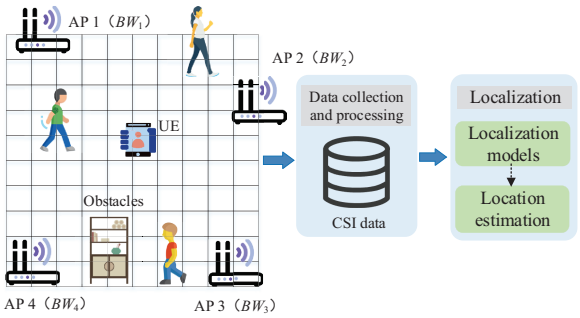


Fig. 1: WiFi localization system model.

verify the superior localization performance of Meta-GLoc.

The rest of this paper is organized as follows. Section II introduces the system model and preliminaries. Section III presents an overview of the proposed Meta-GLoc. Section IV illustrates the experimental settings and evaluates the localization performance of Meta-GLoc, and the whole paper is concluded in Section V.

II. System Model and Motivation

In this section, we first introduce the system model of WiFi localization. Then, we present the preliminaries of meta-learning and GNN, and explain the motivation of adopting them for localization.

A. System Model

As shown in Fig. 1, we consider a WiFi localization system consisting of N WiFi APs. The user equipment (UE) measures the CSI from the packet transmitted by the APs and further utilizes it for localization. Let N_n^{tx} and N^{rx} denote the number of antennas for the n-th WiFi AP and the UE, respectively. The bandwidth of the n-th AP is denoted as BW_n, which is divided into K_n subcarriers due to the adoption of the orthogonal frequency division multiplexing technique. Then, the CSI between the n-th WiFi AP and the UE at the k-th subcarrier can be expressed as

$$\boldsymbol{H}_{n,k} = \sum_{l=1}^{L} \alpha_l e^{-jd_l/\lambda_k} \boldsymbol{a}(\theta_l)\boldsymbol{a}(\phi_l)^H \in \mathbb{C}^{N^{\mathrm{rx}} \times N_n^{\mathrm{tx}}}, \quad (1)$$

where L is the number of transmission paths, α_l is the path attenuation, d_l is the path distance, θ_l is the angle of departure, and ϕ_l is the angle of arrival (AoA) with $\boldsymbol{a}(\cdot)$ being the steering vector. Moreover, λ_k is the wavelength of the k-th subcarrier.

Traditional localization methods (e.g., MUSIC algorithm [14]) aim to extract the AoA and distance of the line-of-sight path from CSI for estimating the UE's location. Therefore, they perform poorly in NLoS scenarios. To overcome this limitation, we consider the DL-based method in this paper.

B. Meta-Learning Technique

Although DL-based methods have the potential to achieve superior performance, neural network models trained in historical scenarios could not adapt well to the new scenario. To address this challenge, the meta-learning technique [6] can be adopted and it only requires some CSI samples collected from the new scenario to fine-tune the model parameters. Specifically, meta-learning mainly includes two stages: meta-training and fine-tuning. The meta-training stage aims to obtain a set of meta-parameters using existing datasets for multiple scenarios, which are used as initial parameters in the fine-tuning stage. Considering the CSI datasets measured from P existing scenarios, the dataset of each scenario $\mathcal{T}_p$ is divided into a support set $\mathcal{S}_p$ for model training and a query set $\mathcal{Q}_p$ for model validation, i.e., $\mathcal{T}_p = [\mathcal{S}_p, \mathcal{Q}_p]$. In the fine-tuning stage for the new scenario, the model is fine-tuned using a few CSI samples measured from the new scenario to achieve adaptive localization.

However, existing meta-learning-based methods only focus on environmental changes while overlooking variations in device configurations. To address this issue, we aim to develop a novel GNN that can adapt to different input sizes.

C. Graph Neural Network

The GNN is a powerful machine learning model for processing graph-structured data [12]. It primarily consists of three modules: graph construction, feature propagation, and prediction output, detailed in the following.

Graph construction. A graph is generally represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, and $\mathcal{X}$ denotes the node features [12]. The neighborhood of a node v is defined as $\mathcal{N}(v) = \{u \in \mathcal{V} | (v,u) \in \mathcal{E}\}$. Assuming N nodes, the connection of the graph can be represented by an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ as

$$\boldsymbol{A}_{n,m} = \begin{cases} w_{n,m}, & \text{if } n \in \mathcal{N}(v), \\ 0, & \text{otherwise}, \end{cases} \quad (2)$$

where $w_{n,m}$ represents the edge feature (i.e., weight) of edge $e_{n,m}$. Moreover, there is a special graph, i.e., undirected graph, and the adjacency matrix is symmetrical.

Feature propagation. After completing the graph construction, we need to learn the feature representation using graph convolutional layers. Considering G graph convolutional layers, the feature of the n-th input node at the g-th layer is $\boldsymbol{s}_n^{(g)}$, with the initial feature given by $\boldsymbol{x}_n^{(0)}$. The detailed steps of feature propagation are as follows.

Message passing. At the g-th layer, the feature of v_n and all its adjacent nodes in the $(g-1)$-th layer and the edge feature $w_{n,m}$ are used to generate messages to update the feature of v_n. The message can be given by

$$\boldsymbol{s}_{n,m}^{(g)} = \varphi^{\mathrm{M},(g)}(\boldsymbol{s}_n^{(g-1)}, \boldsymbol{s}_m^{(g-1)}, w_{n,m}), \quad (3)$$

where $\varphi^{\mathrm{M},(g)}(\cdot)$ is a message function.

Aggregation. Based on the generated message, the node v_n aggregates the messages from all its neighboring nodes, as

$$\hat{\boldsymbol{s}}_n^{(g)} = \underset{m \in \mathcal{N}(n)}{\mathrm{Agg}} (\boldsymbol{s}_{n,m}^{(g)}), \quad (4)$$

where $\mathrm{Agg}(\cdot)$ is an aggregation function, such as a summation function, with variable input dimensions.

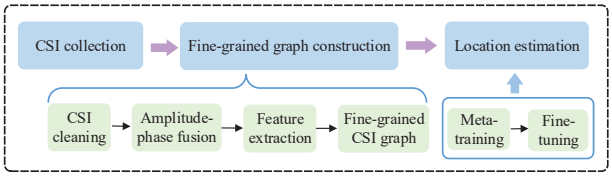


Fig. 2: The framework of Meta-GLoc.

Feature update. The feature of v_n at the g-th layer can be updated by merging the $\boldsymbol{s}_n^{(g-1)}$ and $\hat{\boldsymbol{s}}_n^{(g)}$, as

$$\boldsymbol{s}_n^{(g)} = \varphi^{\mathrm{U},(g)}(\boldsymbol{s}_n^{(g-1)}, \hat{\boldsymbol{s}}_n^{(g)}), \tag{5}$$

where $\varphi^{\mathrm{U},(g)}(\cdot)$ is an update function.

Prediction output. After G layers, the updated node features are $\{\boldsymbol{s}_n^{(G)}\}_{n=1}^{N}$ for each graph. Subsequently, the features are processed through the pooling module (e.g., global average pooling) and fed into the fully connected layer to output the predicted location.

As analyzed above, GNN has a dynamic topological structure that can cope with changes in the number of APs. However, two key challenges remain unresolved. First, the amplitude and phase of the CSI are affected by different sources of noise and error. Directly feeding the raw CSI into the GNN may degrade the localization performance. Second, the bandwidth of each AP may fluctuate dynamically, and the number of antennas varies across APs. This leads to a change in the size of the CSI, making it challenging to input the data into the GNN which requires consistent data dimensions for each node during processing. Therefore, we propose Meta-GLoc to solve the above problems with GNN and meta-learning.

III. The Proposed Meta-GLoc

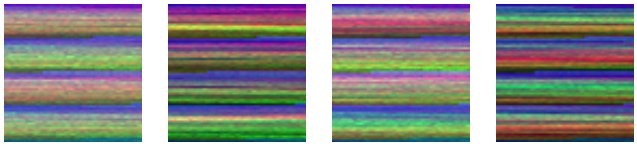
In this section, we first describe the overview of Meta-GLoc. Next, we introduce how the amplitude and phase are processed to construct fine-grained CSI graphs to enhance the adaptability and robustness of Meta-GLoc. Finally, the training strategy based on meta-learning is presented.

A. System Overview

As depicted in Fig. 2, Meta-GLoc consists of three parts: CSI collection, fine-grained GNN construction, and location estimation. First of all, Meta-GLoc collects CSI samples in existing P indoor scenarios to construct the localization dataset, and a proposed method is adopted to clean the raw amplitude and phase, removing the noise and other error components. Then, an amplitude-phase fusion method based on flip techniques and a feature extraction method based on spatial pyramid pooling are designed to construct fine-grained CSI graphs. With the graph, GNN is further employed to realize localization. Moreover, to support the cross-domain localization, Meta-GLoc also contains a training strategy, consisting of two stages: meta-training and fine-tuning.

(a) R. (b) B. (c) G. (d) RGB.

Fig. 3: The CSI image for different channels at location 1.



(a) Loc. 1. (b) Loc. 2. (c) Loc. 3. (d) Loc. 4.

Fig. 4: CSI images measured from four different locations.

B. Amplitude-Phase Fusion

Before utilizing the collected CSI for amplitude-phase fusion, it is essential to clean the data to reduce noise and errors caused by hardware imperfections. For amplitude processing, a filtering algorithm [15] based on the discrete wavelet transform is applied to achieve smooth denoising while effectively preserving signal characteristics. For the phase over different subcarriers, there are errors caused by the carrier frequency offset and sampling time offset, and we utilize the sanitization algorithm proposed in [16] to eliminate those offsets.

After cleaning, the next step is to feed the CSI into a GNN for localization. According to Section II-A, each CSI sample consists of four dimensions: the transmit antenna dimension, the receive antenna dimension, the subcarrier dimension, and the dimension representing the complex values constructed from amplitude and phase. Thus, it is not suitable for direct usage since most existing neural networks cannot process such a kind of data structure. To address this issue, we aim to fuse each CSI sample into one CSI image due to the success of image tasks using neural networks [17]. Existing work [8] used CSI of three WiFi APs to construct one image, but this method faces the challenge of variations in the number of APs. Thus, we opt to convert the CSI to an image for each AP.

CSI consists of both amplitude and phase components, and they can serve as two distinct channels within the RGB channels of the CSI image. Specifically, for the amplitude/phase channels, the size of the CSI is $N_n^{\mathrm{tx}} \times N^{\mathrm{rx}} \times K_n$, and we need to reshape it into a two-dimensional matrix with the size being $N_n^{\mathrm{img}} \times N_n^{\mathrm{img}}$, where $N_n^{\mathrm{img}} = \lceil \sqrt{N_n^{\mathrm{tx}} \times N^{\mathrm{rx}} \times K_n} \rceil$ and $\lceil \cdot \rceil$ is a ceiling function. Here, we choose a reshaped dimension close to the CSI size since it can maximize the integrity of the CSI and avoid inserting too much invalid information. For the remaining channel, we choose the amplitude channel as the third one due to its lower susceptibility to hardware imperfections. Moreover, we apply the up-down flipping for data augmentation, since it changes the spatial arrangement of the subcarriers, effectively simulating variations in signal propagation patterns.

Until now, we can show the constructed CSI image with RGB channels in Fig. 3, and its size is $N_n^{\mathrm{img}} \times N_n^{\mathrm{img}} \times 3$. The data in the R, B, and G channels are amplitude, phase, and

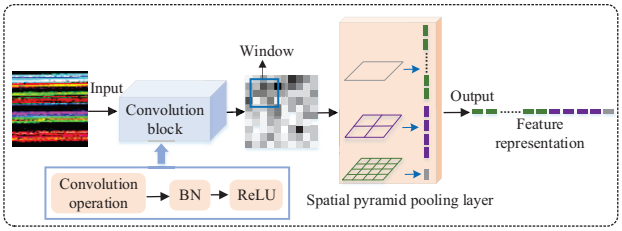


Fig. 5: Feature extraction based on spatial pyramid pooling.

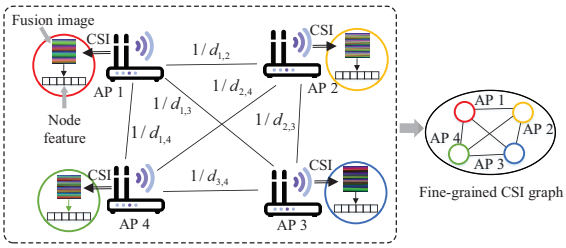


Fig. 6: The fine-grained CSI graph construction method.

amplitude after up-down flipping, respectively. We first convert the data of three channels into grayscale images and then fuse them into an RGB image. Fig. 4 shows the CSI images constructed from four different locations. We can see that they are different from each other, providing unique location features for indoor localization.

C. Feature Extraction

Once the CSI image is constructed, we proceed to extract features. Most of the existing work adopts the CNN directly, but it cannot be used directly in this paper because dynamic AP configurations lead to varying sizes of the constructed images. If the same CNN network is applied, the output feature dimensions differ for different APs. An intuitive solution is to use different CNNs. However, this results in higher training complexity, and feature output consistency among different CNNs is difficult to guarantee. To tackle this problem, we employ the spatial pyramid pooling [13] to design a feature extraction method. As shown in Fig. 5, it consists of convolutional layers with rectified linear units (ReLUs) and batch normalization (BN) layers, a spatial pyramid pooling layer, and a fully connected layer. Specifically, the generated CSI image is first converted into a feature map of size $z \times z$, where z is related to the number of subcarriers K_n.

The CSI feature map is then input into the spatial pyramid pooling layer for down-sampling. It divides the CSI feature map into grid areas of multiple scales (or pyramid levels), performs pooling operations on each area to capture global and local multi-scale information, and generates fixed-length feature representations. To be specific, we consider an L^{pool}-level pyramid. The output feature size at the l-level pyramid is $c_l \times c_l$. To obtain $c_l \times c_l$ features, we perform sliding window pooling on the CSI feature map at each pyramid level. The window size is $\lceil z/c_l \rceil$ and the stride size is $\lfloor z/c_l \rfloor$, where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the ceiling and floor operations, respectively. After pyramid pooling is completed, the fully connected layer concatenates L^{pool} outputs to obtain a fixed-length feature representation. This pooling part reduces the size of the CSI feature map and extracts important features, which helps to reduce the computational complexity and prevent overfitting.

D. Graph Construction and Localization

Until now, we have obtained the feature for each AP. We then employ features of all APs to construct a fine-grained graph as shown in Fig. 6, which can adapt to the varying number of APs. Specifically, as introduced in Section II-C, a graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, which requires both node features and edge features. For N WiFi APs, each AP is regarded as a graph node. The CSI feature, denoted by $\boldsymbol{x}_n^{(0)}$, obtained after amplitude-phase fusion and feature extraction, is used as the graph node feature.

Due to the limited indoor space and the small scale of AP deployment, it is reasonable to assume that any two APs in the indoor scenario are connected in the graph, and the constructed graph is a complete one. To capture the spatial relationship between any two APs and signal propagation characteristics, the weights in the adjacency matrix $\boldsymbol{A}$ should be related to the distance between any two APs. Since each AP node would update its feature using the features from its neighbors and features from far away APs should be naturally suppressed, we consider that the weight is set as the inverse of the distance between APs, and the corresponding $\boldsymbol{A}$ is given by

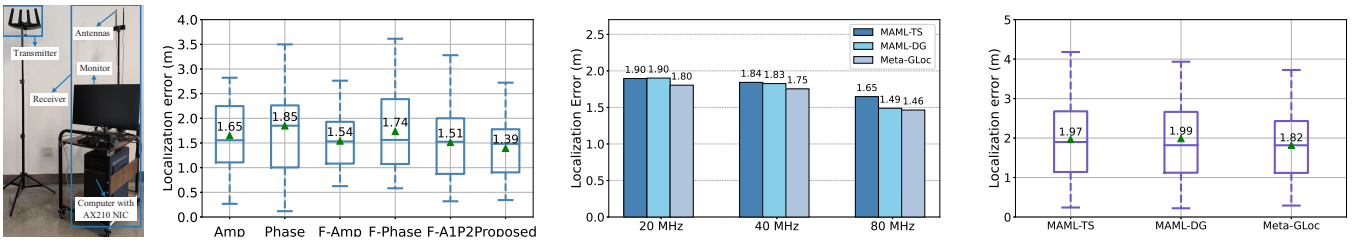
$$\boldsymbol{A}_{n,m} = \begin{cases} 1/d_{n,m}, & \text{if } n \in \mathcal{N}(v), \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where $d_{n,m}$ is the Euclidean distance between the n-th AP and the m-th AP. With equation (6), the spatial perception of the graph is effectively improved.

The constructed fine-grained CSI graph is then input into the GNN for location estimation. Here, we apply three graph convolutional layers to capture the relationships between nodes through a feature propagation mechanism. At each layer, the feature of each node is updated by aggregating information from its neighboring nodes through graph convolution operations. Each graph convolutional layer is followed by a ReLU activation and a BN layer to enhance non-linearity and stabilize training. After the three graph convolutional layers, we employ a global average pooling layer to aggregate the node features into a graph-level representation. This representation is then passed through a fully connected layer to perform accurate location estimation. To train the model, we use the mean square error loss function since the localization task can be regarded as a regression problem. Notably, the GNN's insensitivity to the number of nodes makes it robust against variations in the number of APs.

E. Meta-Learning Training Strategy

In the previous section, we have constructed the fine-grained CSI graphs along with a corresponding GNN, which can adapt to the dynamic AP configuration within the same scenario. However, when deploying the GNN model in the



(a) (b) Different CSI graphs. (c) Bandwidth. (d) Real-world setting.

Fig. 7: (a) Prototype of Meta-GLoc and the localization performance under (b) different CSI graphs, (c) different bandwidths, and (d) a real-world setting.

new scenario, it still requires re-collecting CSI and re-training the model, incurring significant overhead. To address this issue, we integrate the GNN with a meta-learning strategy, which consists of two stages: meta-training and fine-tuning. For clarity, the GNN established in the previous section is denoted by $f_{\boldsymbol{\theta}}$ with $\boldsymbol{\theta}$ being the trainable parameters.

Meta-training. The training samples are collected from P existing scenarios and randomly divided into I^{tr} groups. Each group is further split into a support set $\mathcal{S}_p$ and a query $\mathcal{Q}_p$. The training process consists of an inner loop and an outer loop. The inner loop aims to update the group-specific adaptive parameters on the support set, while the outer loop aims to update the global meta-parameters on the query set.

In the inner loop, we use the gradient descent update to obtain the group-specific parameters $\boldsymbol{\theta}'_p$, which is given by

$$\boldsymbol{\theta}'_p \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}; \mathcal{S}_p), \tag{7}$$

where α is the learning rate in the inner loop and $\mathcal{L}(f_{\boldsymbol{\theta}}; \mathcal{S}_p)$ is the loss over the support set $\mathcal{S}_p$. In the outer loop, the loss over the query sets of I^{tr} groups is calculated as $\sum_{p=1}^{I^{\mathrm{tr}}} \mathcal{L}(f_{\boldsymbol{\theta}'_p}; \mathcal{Q}_p)$. To update this loss, the stochastic gradient descent is adopted, and the meta-parameters are updated as

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{p=1}^{I^{\mathrm{tr}}} \mathcal{L}(f_{\boldsymbol{\theta}'_p}; \mathcal{Q}_p), \tag{8}$$

where β is the learning rate in the outer loop.

Fine-tuning. After completing meta-training, the meta-trained parameters are used as initial parameters for the GNN model in the fine-tuning stage. Specifically, A few CSI samples collected in the new scenario are preprocessed and constructed into the fine-grained graphs, which are then fed into the pre-trained GNN model for fine-tuning following the update formula used in the inner loop. This adaptation enables adaptive localization in the new scenario.

IV. Experiment Evaluation

In this section, we first elaborate on the experiment setup and then provide the experiment results and analysis.

A. Experiment Setup

We conduct experiments in three indoor scenarios: a classroom, a meeting room, and a laboratory. In each scenario, four WiFi APs (i.e., TP-LINK AX5400 routers) are deployed to transmit WiFi signals. As shown in Fig. 7(a), a computer with a two-antenna Intel AX210 network interface card (NIC), with an antenna spacing of 6 cm, acts as the UE. To simulate a real-world scenario, the receiver antenna height is set to 1.55 m, and the antenna height of the WiFi AP is 1.8 m. Both the WiFi APs and Intel AX210 NIC operate under the IEEE 802.11ax standard, with the center frequency being 5.25 GHz and bandwidth randomly distributed within the set of [20, 40, 80] MHz. To collect CSI at the UE, we set the UE to work in the monitor mode and utilize the PicoScenes tool [18], with a sampling frequency of 100 Hz. In each scenario, we set reference points (RPs) with a distance of 0.5 m between two RPs to collect CSI samples. In meta-learning, we adopt some scenarios as the historical ones, with 80 CSI samples of each RP used for training, and the remaining scenario(s) as the new one(s), with 20 CSI samples used for fine-tuning. The software framework of the Meta-GLoc is implemented using Python 3.9.0 and PyTorch 1.11.0.

B. Performance Analysis

In this subsection, we first analyze the effectiveness of the constructed CSI graphs, and then measure the performance under different bandwidths. Finally, we evaluate the performance of Meta-GLoc in a real-world setting.

1) Effectiveness of the constructed CSI graph. To analyze the effectiveness of the proposed CSI graph construction method, we compare it against five intuitive methods: "Amp", "Phase", "F-Amp", "F-Phase", and "F-A1P2". "Amp" and "Phase" utilize the preprocessed amplitude and phase directly to construct CSI graphs. "F-Amp" and "F-Phase" employ the fine-grained graph construction scheme based on amplitude and phase fusion methods, respectively. In the amplitude fusion method, three channels of the CSI image are filled with the amplitude, the up-down flipped amplitude, and the left-right flipped amplitude. The phase fusion method adopts a similar strategy. "F-A1P2" adopts a scheme similar to the proposed scheme. However, unlike our method detailed in Section III-B, it fills two RGB channels with amplitude and phase, while the third channel is filled with the up-down flipped phase. In Fig. 7(b), "▲" represents the mean error.

From Fig. 7(b), methods utilizing amplitude information achieve lower mean localization error compared to those using phase information. This is because the raw phase values

collected by the wireless NIC are inaccurate, introducing additional errors that are difficult to remove completely. Moreover, the mean localization error obtained using the fine-grained graph construction methods is lower than that using the original graph construction methods. This indicates that the proposed fine-grained graph construction method enhances CSI feature representation, thereby improving the localization robustness. Furthermore, our proposed method demonstrates superior localization performance.

2) Performance under different bandwidths. To evaluate localization performance under different bandwidths, we compare Meta-GLoc with the state-of-the-art methods, i.e., MAML-TS and MAML-DG [8]. Here, we train the model using the dataset collected in the classroom with a bandwidth of 80 MHz, and test it using the dataset collected in the presence of obstacles under varying bandwidths. As shown in Fig. 7(c), the mean localization error gradually decreases as the bandwidth increases across the three methods. This is because a larger bandwidth contains more subcarriers, providing richer CSI features that enhance localization performance. Among the three methods, Meta-GLoc achieves the best localization performance across all bandwidth settings. This confirms that Meta-GLoc effectively utilizes the available CSI under different bandwidths, achieving lower localization errors than the other two methods.

3) Real-world experiment. Now, we evaluate the performance of Meta-GLoc in a real-world setting, where both the number of APs and the bandwidth vary dynamically. To assess our method, we compare Meta-GLoc with MAML-TS and MAML-DG. Specifically, we use CSI samples from the classroom and the laboratory for meta-training, and part of the CSI samples from the meeting room for fine-tuning. The fine-tuned model is then tested under a random number of APs with the bandwidth of each AP randomly selected from the set $\{20, 40, 80\}$ MHz. In this setting, Fig. 7(d) presents the localization performance of the methods. We can see that Meta-GLoc achieves the best performance among the three methods. This is because Meta-GLoc can fully utilize the available WiFi APs and extract rich CSI features, whereas MAML-TS and MAML-DG are limited to three APs. Moreover, MAML-TS and MAML-DG do not consider bandwidth variations, leading to lower localization performance.

V. Conclusion

In this paper, we have introduced Meta-GLoc, an adaptive and robust WiFi localization system designed to address the challenges of dynamic AP configurations. First, we have introduced GNNs to effectively leverage the spatial topological information of APs to cope with the variations of APs' location and number. Then, we have proposed a fine-grained CSI graph construction scheme by incorporating an amplitude-phase fusion method and a feature extraction method, thereby improving the localization robustness. The amplitude-phase fusion method exploits CSI in both the frequency and spatial domains, fusing amplitude and phase into CSI images to improve data reliability. Meanwhile, the feature extraction method utilizes spatial pyramid pooling to generate bandwidth-independent and dimension-consistent features, thereby resolving the dimensionality issue. Furthermore, we have combined GNN with meta-learning to improve the cross-scenario generalization. Experiment results have demonstrated that Meta-GLoc effectively tackles dynamic AP configurations, significantly improving localization accuracy and robustness.

References

[1] R. C. Shit, S. Sharma, D. Puthal, and A. Y. Zomaya, "Location of things (LoT): A review and taxonomy of sensors localization in IoT infrastructure," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2028–2061, 3rd Quart. 2018.

[2] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3607–3644, 4th Quart. 2018.

[3] X. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 316–327, Jan. 2020.

[4] A. Foliadis, M. H. C. Garcia, R. A. Stirling-Gallacher, and R. S. Thomä, "CSI-based localization with CNNs exploiting phase information," in *Proc. IEEE WCNC*, Apr. 2021, pp. 1–6.

[5] H. He, H. Hu, X. Huan, H. Liu, J. An, and S. Mao, "AI generated signal for wireless sensing," in *Proc. IEEE GLOBECOM*, Dec. 2023, pp. 6097–6102.

[6] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, Aug. 2017, pp. 1126–1135.

[7] W. Wei, J. Yan, X. Wu, C. Wang, and G. Zhang, "A meta-learning approach for device-free indoor localization," *IEEE Commun. Lett.*, vol. 27, no. 3, pp. 846–850, Mar. 2023.

[8] J. Gao, D. Wu, F. Yin, Q. Kong, L. Xu, and S. Cui, "MetaLoc: Learning to learn wireless localization," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3831–3847, Dec. 2023.

[9] Y. Zhang and K. Psounis, "Efficient indoor localization via switched-beam antennas," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2101–2115, Sep. 2020.

[10] 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.*, vol. 9, no. 4, pp. 2378–2391, Jul. 2022.

[11] M. Stahlke, G. Yammine, T. Feigl, B. M. Eskofier, and C. Mutschler, "Indoor localization with robust global channel charting: A time-distance-based approach," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 1, pp. 3–17, Mar. 2023.

[12] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[14] Y. Zheng, M. Sheng, J. Liu, and J. Li, "Exploiting AoA estimation accuracy for indoor localization: A weighted AoA-based approach," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 65–68, Feb. 2019.

[15] S.-H. Fang, W.-H. Chang, Y. Tsao, H.-C. Shih, and C. Wang, "Channel state reconstruction using multilevel discrete wavelet transform for improved fingerprinting-based indoor localization," *IEEE Sensors J.*, vol. 16, no. 21, pp. 7784–7791, Nov. 2016.

[16] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using WiFi," in *Proc. ACM SIGCOMM*, Aug. 2015, pp. 269–282.

[17] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern Recognit.*, vol. 35, no. 10, pp. 2279–2301, Oct. 2002.

[18] Z. Jiang, T. H. Luan, X. Ren, D. Lv, H. Hao, J. Wang, K. Zhao, W. Xi, Y. Xu, and R. Li, "Eliminating the barriers: Demystifying Wi-Fi baseband design and introducing the PicoScenes Wi-Fi sensing platform," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4476–4496, Mar. 2022.