

Meta-Learning for Beam Prediction in a Dual-band Communication System

Ruming Yang, *Student Member, IEEE*, Zhengming Zhang, *Student Member, IEEE*, Xiangyu Zhang, *Student Member, IEEE*, CHunguo Li, *Senior Member, IEEE*, Yongming Huang, *Senior Member, IEEE*, Luxi Yang, *Senior Member, IEEE*

Abstract—Large antenna arrays and beamforming are necessary for the mmWave communication system, resulting in heavy time and energy consumption in the beam training stage. Therefore, dual-band operations are expected to be deployed in future communication systems, where low-frequency channels are used to meet basic communication needs, and millimeter wave (mmWave) channels are exploited when the high-rate transmission is required. Existing works utilize deep learning methods to extract low-frequency channel state information (CSI) to reduce the mmWave beam training overheads. However, an important limitation of deep learning approaches is that the model is usually trained in a given environment. When employed in an unseen environment, it usually requires a large amount of data to retrain. In this paper, a model-agnostic optimization algorithm based on meta-learning is proposed to provide a general mmWave beam prediction model. This model can be deployed to edge base stations and effectively adapted to the environment without the need for a heavy collection of data. Simulation results demonstrate that the proposed approach could reduce the model adaptation overheads. The meta-learning-based beam prediction model is robust and achieves high prediction accuracy and spectral efficiency in different signal-to-noise ratio (SNR) regimes.

Index Terms—Dual-band, mmWave, beam prediction, meta-learning, machine learning,

I. INTRODUCTION

MILLIMETER wave (mmWave) communication is one of the key technologies in the fifth-generation (5G) wireless communications. Thanks to its large available bandwidth, a high transmission rate can be achieved to meet the ever-growing demand in different scenarios. Due to the high sensitivity to blockage of mmWave bands, mmWave communication technologies are usually designed in the indoor environment. For outdoor communications, large antenna arrays and beamforming are considered to offer a high directional link between the base station (BS) and user equipment (UE). As the

This work was supported by the National Key Research and Development Program of China under Grant 2020YFB1804901 and the National Natural Science Foundation of China under Grants 61971128, U1936201, 62225107, and the Fundamental Research Funds for the Central Universities 2242022k30002. (Corresponding author: Luxi Yang)

R. Yang, Z. Zhang, X. Zhang, Y. Huang, and L. Yang are with the National Mobile Communications Research Laboratory, Frontiers Science Center for Mobile Information Communication and Security, School of Information Science and Engineering, Southeast University, Nanjing 210096, China, and also with Purple Mountain Laboratories, Nanjing 211111, China (e-mail: yagnrm@seu.edu.cn; zmzhang@seu.edu.cn; xy_zhang@seu.edu.cn; huangym@seu.edu.cn; lxyang@seu.edu.cn).

C. Li are with the National Mobile Communications Research Laboratory, School of Information Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: chunguo@seu.edu.cn).

high directional beam has a narrow bandwidth, beam training is required at both BS and UE sides to achieve stable beam alignment [1]–[3]. Traditional methods adopt exhaustive or hierarchical searching over the beam codebook [4], [5]. Since the BS has to estimate the channel before beam selection, channel state information sent from UE is generally required. BS and UE have to frequently interact during beam alignment, which results in a high time and energy overheads.

To overcome the challenges of mmWave band, future communication systems are expected to operate in multiple bands [6]–[9]. BSs and UEs utilize the low-frequency channels to keep in connection and communicate in the mmWave channels when a high data rate is required. Existing works focus on analyzing the feasibility of using the information of low-frequency channels to solve some problems in mmWave communications. The author in [7] demonstrates that the sub-6GHz and mmWave bands have substantially similar power azimuth spectrums (PAS). In paper [8], the author exploits sub-6GHz vehicle to everything (V2X) communications to design a medium access control (MAC) for mmWave vehicular communications. In particular, they propose to decouple the control and data planes and use sub-6GHz broadcast omnidirectional V2X communications for the control plane and directional mmWave communications for the data plane. In paper [9], the author investigates the feasibility of conducting a coarse angle of arrival (AoA) estimation on the sub-6GHz channel and then utilizes the fully-analog beamforming for fine-tuning and transmissions. Based on the estimated sub-6GHz AoA, the angular range scan for the mmWave transmitter is greatly reduced.

In order to efficiently extract information in the sub-6GHz bands, researchers have applied deep learning [10]–[13] to resolve the beam selection problem [14]–[16] in dual-band communication systems [17]–[19]. In paper [19], the author takes a complex sub-6GHz channel of each subcarrier as the input of a deep neural network (DNN) and predicts the optimal mmWave beam. The author in paper [18] regards the complex sub-6GHz channel data as input and develops a 3-dimensional (3D) convolutional neural network (CNN) to provide the solution. Regrettably, these deep learning-based approaches will also bring two main problems. The first is that it is hard for the deep learning model learned from simulation data to work in the time-varying channel environment. Transfer learning methods have to be utilized to retrain the model. However, the scarcity of high-quality labeled data from realistic wireless communication systems makes

a model that needs to retrain with heavy overheads of data collection unreasonable. The second is the deep learning model usually designed for one specific communication environment with a fixed topology. When topology changes, such as the location of BS or the antennas' orientation, the performance of the deep learning methods will be significantly affected. Thus to enable deep learning in dual-band communications, it is vital to design a general model that can be easily adapted to realistic environments and robust to various topologies.

Meta-learning [20] is currently under intensive investigation in the field of deep learning, which aims to tackle the few-shot learning and deep transfer learning problems. Meta-learning is good at generalizing the knowledge from past training tasks and transferring it to unseen new tasks. Prior works explored from three directions: (1) model parameter initialization [21]–[24]. These approaches try to find a better initialization model compared with traditional methods. Good initialization can greatly reduce the cost of training and speed up convergence. (2) Metric-based mapping function [25]–[28]. This kind of method focuses on exploring the relationship between the known knowledge and the novel samples. (3) Meta-model-optimizer [29]–[32]. Approaches based on this direction mainly learns to generate a more efficient optimizer. Applications of meta-learning to communication systems are gradually increasing, including channel estimation [33]–[35], edge caching [36] and wireless sensor networks [37], etc.

This paper proposed a meta-learning-based adaptive beam prediction algorithm to provide an easy-transfer model for sub-6GHz and mmWave dual-band communication systems. We summarize our contributions as follows:

(1) To the best of our knowledge, we are the first to investigate a meta-learning-based model for the few-shot transfer learning problem in the sub-6GHz and mmWave dual-band communication system. Concretely, we propose a beam prediction method that takes in channel state information (CSI) at the sub-6GHz band and outputs the optimal mmWave beam. The whole scheme is valid for various wireless communication environments, which can be deployed to edge base stations and effectively personalized without the high overheads of transferring.

(2) We design a meta-learner network based on the meta-learning paradigm and train it to learn a well-generalized initialization for potential tasks. A bi-level meta-learning structure, including inner loop and outer loop, is designed to optimize the meta-learner. Specifically, we separate the traditional multiple layer perception (MLP) network into two parts: the encoder and the decoder parts. In the training phase, only the encoder part is optimized in the inner loop, which can slightly reduce the overheads of computing the second derivative. In the adaptation phase, only the decoder part is updated, and the computational overheads of model transfer is greatly reduced. Theoretical analysis proves that the proposed meta-learner pays more attention to the inner product of gradient optimization directions between iterations than the traditional learner based on gradient descent, which significantly improves the model's generalization ability.

(3) We evaluate our meta-learning algorithm on the publicly available dataset DeepMIMO [38]. This dataset generates

wireless channels with accurate 3D ray-tracing. Simulation results demonstrate that the proposed meta-learning-based algorithm achieves higher prediction accuracy and spectral efficiency in different signal-to-noise rate (SNR) regimes. We utilize the t-distributed stochastic neighbor embedding (t-SNE) [39] to visualize the feature embeddings of our algorithm. The result indicates that our algorithm is more discriminant than the direct transfer learning method. We also conduct simulations to investigate the impact of gradsteps and number of samples. The results help us to choose the appropriate hyperparameters for adaptation.

The rest of this paper is organized as follows. The system model and beam prediction problem formulation parts will be introduced in Section II. The deep transfer learning problem is formulated in Section III. Section IV explains the proposed meta-learning-based beam prediction method, and the theoretical analysis is conducted in Section V. Results of ray tracing-based simulations are given in Section VI. Finally, the conclusion is presented in Section VII.

Notation: We use the following notation throughout this paper: \mathbf{A} is a matrix, \mathbf{a} is a vector, a is a scalar, \mathcal{A} is a set of scalars, and \mathbb{A} is a set of vectors. \mathbf{I} is the identity matrix. $\mathcal{CN}(m; \mathbf{R})$ is a complex Gaussian random vector with mean m and covariance \mathbf{R} .

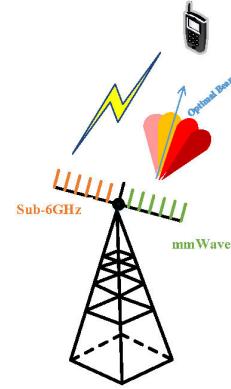


Fig. 1: Schematic diagram for the system model with the sub-6GHz and mmWave bands. Note that BS employs both sub-6GHz antennas and mmWave antenna arrays. UE with a single antenna can be attached in both bands.

II. SYSTEM MODEL AND BEAM PREDICTION

In this section, we explain our dual-bands communication system model and formulate the beam prediction problem. Considering the communication system model in Fig. 1, the BS is equipped with M_{sub} sub-6GHz antennas and an M_{mmW} -elements mmWave antenna arrays. The UE with a single antenna can be attached in both bands. In this work, the proposed beam prediction can be considered as an initial link establishment, no mobility is considered during the beam prediction. Therefore, in the beginning time slot, the channels can be simply regard as the point-to-point channels. Next, we explain the channel model in detail and formulate the beam prediction problem.

A. System Model

We consider a multiple-input single-output (MISO) channel that BSs are equipped with uniform linear arrays (ULAs) and UEs with isotropic antennas [19]. We assume the sub-6GHz transceiver adopts a fully-digital architecture, and let $\mathbf{h}_k^{sub} \in \mathbb{C}^{M_{sub} \times 1}$ denotes the uplink channel vector of the k -th subcarriers from UE to BS, where $k = 1, \dots, K$ is the index of subcarriers and M_{sub} is the number of sub-6Hz antenna of the BS side. Then the uplink signal received by the BS can be expressed as

$$\mathbf{y}_k^{sub} = \mathbf{h}_k^{sub} s_k + \mathbf{n}_k^{sub}, \quad (1)$$

where s_k is the uplink pilot signal, $\mathbf{n}_k^{sub} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I})$ is the additive white Gaussian noise (AWGN) and \mathbf{I} is the identity matrix.

For the downlink transmission, we assume that the BS works in the mmWave bands and only uses analog beamforming technology with M_{mmW} phase shifters. Let $\mathbf{f} \in \mathbb{C}^{M_{mmW} \times 1}$ denotes the beamforming vector, which is selected from the codebook \mathbf{F} . The received signal at the UE can be written as

$$\mathbf{y}_k^{mmW} = \mathbf{h}_k^{mmW} \mathbf{f} s_k + \mathbf{n}_k^{mmW}, \quad (2)$$

where $\mathbf{n}_k^{mmW} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I})$ is the AWGN.

B. Channel Model

For the channel model, we employ a geometric (physical) channel presented in [40] and utilize the accuracy 3D ray tracing to generate the parameters of the channel model. The mmWave or sub-6GHz channel model can be expressed as

$$\mathbf{h}_k = \sum_{d=0}^{D_c-1} \sum_{l=0}^L \alpha_l e^{-j \frac{2\pi k}{N} d} p(dT_S - \tau_l) \mathbf{a}(\theta_l, \phi_l), \quad (3)$$

where T_S is the sampling period, τ_l is the delay and D_c is the length of the cyclic prefix. α_l is the path gains, θ_l and ϕ_l are the azimuth angle of arrival (AoA) and zenith angle of arrival (ZoA) of the l -th path, respectively.

C. Beam Prediction

In the 5G new radio (NR) standard, BS and UE will transmit a set of pre-defined beams respectively to search the optimal beam pair, which called beam sweeping. For simplicity, we only consider the beam selection on the downlink mmWave channel, whereas the same method can be easily extended at UE side. Considering the mmWave channel model \mathbf{h}_k^{mmW} , the signal to noise rate (SNR) of the UE at k -th subcarrier can be obtained as

$$\text{SNR}_k = |\mathbf{h}_k^{mmW} \mathbf{f}|^2 / \sigma^2. \quad (4)$$

Then the downlink achievable rate can be written as

$$R = \sum_{k=1}^K \log_2(1 + \text{SNR}_k). \quad (5)$$

We consider a quantized codebook \mathbf{F} which has M_{mmW} different beams. Thus, finding the optimal beam \mathbf{f}^* can be expressed as

$$\mathbf{f}^* = \arg \max_{\mathbf{f} \in \mathbf{F}} R. \quad (6)$$

Since beam \mathbf{f} is selected from \mathbf{F} , finding the optimal beam \mathbf{f}^* is a non-convex problem. Conventional beam prediction method in the 5G NR standard is beam sweeping, which exhaustively searches among a set of pre-defined beams. Because time synchronization must be considered, the larger the beam codebook size, the higher the time overheads of beam sweeping. Utilizing the deep learning method to reduce the search space of beam has been adopted in several works [11], [12], [41], [16].

III. PROBLEM FORMULATION

In this section, we first formulate the beam prediction as a multi-classification problem and employ the deep neural network to give a solution. Then, we consider the deployment problem of the deep beam prediction model in a various communication system as a meta-learning problem and clarify the destination of optimization.

A. Deep Learning Assisted Beam Prediction

Since both low-frequency channel and mmWave channel share the similar spatial propagation characteristics, the uplink channel knowledge at sub-6GHz band can be utilized to estimate the downlink mmWave channel. Follow the proof in [19], when the position-to-channel mapping is bijective, there exists a deterministic mapping between the downlink and the uplink channels, which can be defined as

$$\Phi_{sub-6 \rightarrow mmW} : \mathbf{h}^{sub} \rightarrow \mathbf{h}^{mmW}, \quad (7)$$

where $\mathbf{h}^{sub} \in \mathbb{S}^{sub-6}$ and $\mathbf{h}^{mmW} \in \mathbb{S}^{mmW}$. \mathbb{S}^{sub-6} and \mathbb{S}^{mmW} are the sets of all candidate sub-6GHz and mmWave channels respectively. We use deep neural network to approximate the uplink channel \mathbf{h}^{sub} to downlink channel \mathbf{h}^{mmW} mapping function, then reduce the overheads of beam training in mmWave communications. Due to that DNN only accept the real value data for input, we split the channel \mathbf{h}^{sub} into the real part $\Re(\mathbf{h}^{sub})$ and the imaginary part $\Im(\mathbf{h}^{sub})$ and stack them together,

$$\mathbf{h}^{sub} \rightarrow (\Re(\mathbf{h}^{sub}), \Im(\mathbf{h}^{sub}))^T. \quad (8)$$

Considering the codebook \mathbf{F} is quantized, the mmWave beam prediction problem can be formulated as a supervised multiple-classification task, where each category corresponds to a specific mmWave beam vector \mathbf{f} . Let us denote the parameters of DNN as ϕ , the output of DNN, written as $\text{NET}(\mathbf{h}^{sub}, \phi)$, should contains the conditional probability distribution of the candidate beams. Thus the prediction model can be denoted as

$$\{\mathcal{P}(\mathbf{f}_i)\} = \text{NET}(\mathbf{h}^{sub}, \phi), \hat{\mathbf{f}}_i \in \mathbf{F}, \quad (9)$$

where $\mathcal{P}(\cdot)$ is the probability.

Enabling the DNN model in mmWave communication systems usually operates in two phases, i.e., the offline training phase and the online deploying phase. As described in [19], in the offline training phase, one uplink sub-6GHz pilot is sent to the BS, and a search over the beams of the codebook \mathbf{F} is done for the mmWave downlink during every coherence time. Therefore, at every coherence time, the uplink channel

\mathbf{h}^{sub} and the index n_k of optimal beam \mathbf{f}_k^* for user k can be generated as a data point. When enough data is collected, we use the dataset to train the DNN model. It must denote that the amount of data required to train a DNN fully increases with the depth and width of the DNN. Insufficient data often caused the model to fail to converge. In the deploying phase, the DNN model can be deployed at the BS side, even the cloud. The BS can utilize the DNN model to predict the optimal beam according to the uplink signal without beam training.

Note that the proposed deep learning-based beam prediction model can be embedded in the realistic mmWave communications with out affecting the traditional communication operation. The data collection process is completed offline, does not occupy the communication resources of the system, and will not affect the quality of service. However, due to the changeable communication environment, the applicability of the DNN model will be impacted. Thus, for better performance, after the DNN model is deployed, it is necessary to update with data in the realistic environment.

B. Meta-Learning-Based Model Generalization

As mentioned in Section II, training a DNN model needs plenty of data-label pairs, which means a heavy overheads for data collection and labeling. Moreover, a neural network fully trained for a given task usually has weak adaptability for unexpected perturbations or unseen environments. Applying a DNN model in practice often needs to collect many samples for transfer learning or fine-tuning, which is not reasonable. Therefore, we utilize deep learning to obtain a reliable general model not only in the simulations but also in practice wireless communication systems. This model does not need to collect a lot of data and consume plenty of computing power for updating in a new environment.

Meta-learning is a promising method to solve the problems of difficult model transfer and high transfer learning requirements. Model-agnostic meta-learning (MAML) [22] is one of the most influential meta-learning paradigms. Its primary purpose is to learn a general model by leveraging previous experiences across a set of tasks. Initialized with this general model, the DNN will be easy to fine-tune to new tasks using a limited amount of data. It should be emphasized that model initialization is one of the most crucial skills in deep learning. A good initialization can accelerate model convergence.

In this paper, we use the meta-learning paradigm to learn a general beam prediction model at the cloud, which can be deployed to each edge BS. The BS only needs a small number of data to update the model. Considering that an important difference between meta-learning and transfer learning is the sampling of the source tasks or source domains, we generate data from various source tasks. Assuming there are M_{BS} known BSs and N_{BS} unknown BSs, each BS servers UEs within its range. M_{BS} known BSs constitute the source domains $\{\mathcal{D}_S(m)\}_{m=1}^{M_{BS}}$, the beam prediction task corresponding to each BS constitute the source tasks $\{\mathcal{T}_S(m)\}_{m=1}^{M_{BS}}$. Similarly, N_{BS} unknown BSs constitute the target domains $\{\mathcal{D}_T(n)\}_{n=1}^{N_{BS}}$, the beam prediction task corresponding to each BS constitute the target tasks $\{\mathcal{T}_T(n)\}_{n=1}^{N_{BS}}$. Note that

$\mathcal{D}_S(m) \neq \mathcal{D}_T(n)$ and $\mathcal{T}_S(m) \neq \mathcal{T}_T(n)$ must be guaranteed for all the BS. In this paper, the model transfer is not limited in single-source domain to single-target domain, but extended to the multi-source domains to multi-target domains.

Assuming training a DNN model with multi-source tasks can be expressed as

$$\phi \leftarrow \text{NET}\left(\sum_{m=1}^{M_{BS}} \mathcal{T}_S(m)\right), \quad (10)$$

where ϕ is the parameters of the DNN model. Let ϕ_n^{Opt} , where $n = 1, 2, \dots, N_{BS}$, denotes the optimal prediction model for n -th unknown BS. We want to find ϕ such that the distance $dis(\phi, \phi_n^{Opt})$ is as small as possible for each target task. Therefore, training a general beam prediction model can be formulated as

$$\text{minimize} \sum_{n=1}^{N_{BS}} \left[\frac{1}{2} dis(\phi, \phi_n^{Opt})^2 \right] \quad (11)$$

It is important to note that even formulate the beam prediction as a meta-learning problem, some points still need to be attended to. The first is that the purpose of meta-learning and transfer learning are different. Transfer learning focus on adjusting an existing model to a specific target task, so as to avoid the problems of overfitting and poor convergence that are easy to encounter when training the target task from scratch. The purpose of meta-learning is using existing source tasks to learn a common model that is easier to transfer when facing multiple unknown target tasks. The second is that meta-learning emphasizes learning to learn or continuous learning at a small cost, and the main exploration direction is few-shot learning even zero-shot learning. How much data is necessary to deploy meta-learning paradigm in the mmWave communication system is a meaningful direction at present. To answer these questions, we propose a meta-learning-based beam prediction algorithm in section IV and conduct related experiments in section VI.

IV. MBBP: AN META-LEARNING-BASED BEAM PREDICTION SOLUTION

In this section, we introduce our meta-learning-based beam prediction solution, from now on, we call it MBBP. Firstly, the architecture of the network adopted in the training process and the generation of dataset are illustrated in details. Then an overview of MBBP is presented. Finally, we introduce how to avoid meta-overfitting.

A. Network Architecture

A MLP network is designed as the backbone shown in Fig. 2(a), which has four fully connected layers and one output layer. The first fully connected layer with 1024 neurons followed with a rectified linear unit (ReLU) is adopted as the input layer. Each of the following three fully connected layers has 2048 neurons and followed with ReLU. Dropout is adopted after all ReLUs for alleviating overfitting, and the dropout rate is 0.5. The number of neurons in the output layer depends on the number of classification categories, i.e., the size of the

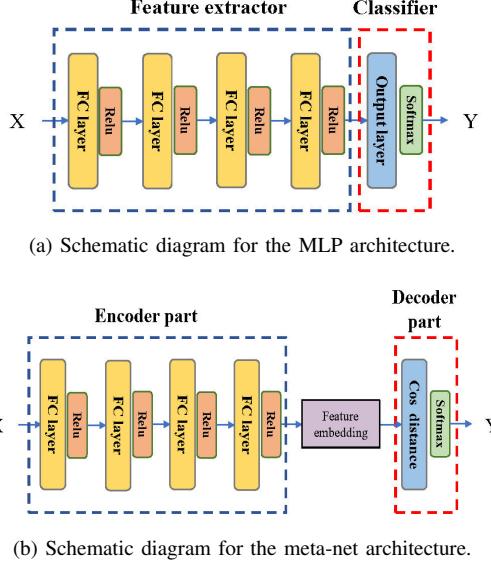


Fig. 2: Illustration of network structure used in the basic training and meta training.

codebook. We adopt softmax after the output layer to obtain each beam's probability.

As we have formulated the beam prediction problem as a meta-learning problem in section III, different base station environments can be regarded as different target domains. One of the major problems of meta-learning algorithm lies in extracting the feature offset between different target domains [42]. In paper [43], the authors also found that, the feature representation output of the feature extraction part before and after adaption phase has a high similarity. However, the output of the head part before and after adaption varies greatly. This suggests that these hidden layers do not change much during adaption, but mostly perform feature reuse. Motivated by these factors, we separate the basic net into two parts, namely the encoder part and the decoder part, as shown in Fig. 2(b). We note this network as the meta-net in the subsequent. The encoder part works as a generic module for extracting sub-6GHz CSI information across subtasks. The decoder part designed as a task-specific output layer for outputting the prediction for the current subtask. In the decoder part, a cosine distances layer is utilized as the output layer. Denote the weight matrix of the cosine distances layer as $\mathbf{W}_{cos} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_C]$, where C is the number of classes. The cosine distances layer is designed to compute the cosine similarity between the input feature embedding \mathbf{E} and each weight vector in \mathbf{W}_{cos} . Note that applying the cosine distance layer here is not our contribution. It is often adopted in prior meta-learning or few-shot learning algorithms [28], [31], [32].

B. Dataset Preparation

The dataset is generated from the DeepMIMO dataset which is constructed using the accurate 3D ray-tracing software. We first selected M_{BS} BSs as the known BS and N_{BS} BSs as the unknown BS to generate corresponding source data sets $\{\mathcal{T}_m^S\}_{m=1}^{M_{BS}}$ and target data sets $\{\mathcal{T}_n^T\}_{n=1}^{N_{BS}}$ respectively.

Algorithm 1 basic learning for beam prediction

Input: Source tasks: $\{\mathcal{T}_m^S\}_{m=1}^{M_{BS}}$, basic training rate: α , number of epochs for basic training: G_B ,
Output: basic net parameter: θ

- 1: Randomly initialize θ
- 2: Randomly select one task \mathcal{T}_B from $\{\mathcal{T}_m^S\}_{m=1}^{M_{BS}}$
- 3: **for** $g = 1, \dots, G_B$ **do**
- 4: sample training batch data from \mathcal{T}_B
- 5: evaluate $\nabla_\theta L(\theta)$ using training batch data
- 6: update θ' : $\theta' = ADAM(\theta, L, \alpha)$
- 7: **end for**

As mentioned in Section III-A, beam prediction for j -th BS can be formulated as a multiple classification task \mathcal{T}_j , where $j \in \{1, 2, \dots, M_{BS} \text{ (or } N_{BS}\)}$. For each \mathcal{T}_j , two datasets should be obtained for meta-training, namely the support dataset \mathcal{D}_{Spt} and query dataset \mathcal{D}_{Qry} . Then, we take the corresponding support datasets $\{\mathcal{D}_{Spt}^S(m)\}_{m=1}^{M_{BS}}$ and query datasets $\{\mathcal{D}_{Qry}^S(m)\}_{m=1}^{M_{BS}}$ for meta-training, $\{\mathcal{D}_{Spt}^T(n)\}_{n=1}^{N_{BS}}$ for fine-tuning and $\{\mathcal{D}_{Qry}^T(n)\}_{n=1}^{N_{BS}}$ for testing. Note that for each task, $\mathcal{D}_{Spt} \cap \mathcal{D}_{Qry} = \emptyset$ should be satisfied.

C. Network Training

Training the MBBP has three stages, namely, the basic training, the meta-training and the fine-tuning stage.

1) *Basic Training*: The basic training stage is designed to train an initial feature embedding model for subsequent meta-training. Initialize with this model can avoid the cold start of meta-training, thereby accelerating the extraction of meta-knowledge and convergence during meta-training. Therefore, in the basic training stage, we utilize the basic net and randomly select one task \mathcal{T}_B from source tasks $\{\mathcal{T}_m^S\}_{m=1}^{M_{BS}}$ as the training task. Then we generate the corresponding training data pairs $\{\mathbf{x}_b, \mathbf{y}_b\}_{b=1}^{N_B}$ as described in section III-A, here N_B is the scale of data in \mathcal{T}_B . The basic net is trained to minimize the cross-entropy loss L and updated via the ADAM [44] algorithm with learning rate α . The cross-entropy loss is defined as

$$L = - \sum_{b=1}^{N_B} \mathbf{y}_b \log_2 (\text{NET}(\mathbf{x}_b, \phi)), \quad (12)$$

where \mathbf{y}_b is the one-hot vector of the b -th label, \mathbf{x}_b is the b -th data point. The basic training algorithm is sketched in Algorithm 1.

2) *Meta-Training*: In the meta-training stage, we adopt the meta-net and initialize the encoder part $f_{enc}(\theta_e)$ with the pre-trained model. Then we design two iterative training processes, i.e., the inner-task training and the across-task training. In the inner-task training process, a task \mathcal{T}_m^S and its corresponding support dataset $\mathcal{D}_{Spt}^S(m)$ and query dataset $\mathcal{D}_{Qry}^S(m)$ are randomly selected from $\{\mathcal{T}_m^S\}_{m=1}^{M_{BS}}$. Then the feature embeddings $\mathbf{E}_{Spt}^S(m)$ of $\mathcal{D}_{Spt}^S(m)$ are generated through the encoder

part. The weights of cosine layer in decoder part $f_{dec}(\theta_d)$ is computed with prototypical feature averaging

$$\mathbf{w}_i = \frac{1}{S_c} \sum_{x_c \in S_c} \mathbf{E}(x_c), \quad (13)$$

where S_c is the c -th class in $\mathcal{D}_{Spt}^S(m)$, and x_c is the data point in the c -th class. The mean computation is to achieve each clusters' anchor point, which can be regarded as the weight vector \mathbf{w}_i of each class. After that, the parameter of $f_{enc}(\theta_e)$ is updated with G_m steps of gradient descents, i.e.,

$$\theta_e^{g+1} = \theta_e^g - \alpha_g \nabla_{\theta_e^g} L(f_{dec}(\mathbf{E}_{Spt}^S(m), \theta_d^g), \mathbf{Y}_{Spt}(m)), \quad (14)$$

where $\{\alpha_g\}_{g=0}^{G_m}$ is a set of gradient step sizes optimized jointly with θ_e .

In the across-task training process, the model is trained towards the optimal of the multi-task joint training. Thus, the meta-net is trained with $\mathcal{D}_{Qry}^S(m)$ from different \mathcal{T}_m in each epoch. The parameters θ_e and θ_d are updated via the ADAM algorithm with across-task learning rate γ , i.e.,

$$\theta^{t+1} = ADAM(\theta^t, \sum_{m=1}^{M_{BS}} L(f_{dec}(\mathbf{E}_{Qry}^S(m), \theta^t), \mathbf{Y}_{Qry}(m)), \gamma), \quad (15)$$

here θ^{t+1} refers to the parameters θ_e^{t+1} or θ_d^{t+1} after $(t+1)$ -th out loop iteration.

3) *Fine-tuning*: In the meta adaption stage, the meta-net updates parameters using the adaption dataset \mathcal{D}_{Spt}^T and tests on the dataset \mathcal{D}_{Qry}^T from a target task $\{\mathcal{T}_n\}_{n=1}^{N_{BS}}$. The updating step can be done in a quite short time because we initialize the encoder part and decoder part with previous parameters θ_e^* and θ_d^* respectively, which is easy to converge.

In the adaption step, the decoder part is updated in G_a steps of gradient descents by

$$\theta_d^{a+1} = \theta_d^a - \alpha \nabla_{\theta_d} L(f_{dec}(\mathbf{E}_{Spt}^T, \theta_d^a), \mathbf{Y}_{Spt}^T), \quad (16)$$

where, \mathbf{Y}_{Spt}^T is the one-hot label of the data in adaption dataset. After adaption is finished, the \mathcal{D}_{Qry}^T is put into the network to get the prediction.

In summary, the whole meta-learning algorithm is sketched in Algorithm 2. It must notes that, the MBBP can be further extended with the deep reinforcement learning, which will be an interesting work in the future.

D. Meta-overfitting

Considering one situation that the model parameters does not move to the minimum of the global tasks during the training process, but moved to the minimum of one local sub-task. We regard this phenomenon as meta-overfitting. Simply applying meta-learning paradigm to the beam prediction problem can not avoid meta-overfitting. Therefore we further utilize an interesting skill to solve this problem.

Assuming that all data points in the source tasks $\{\mathcal{T}_m^S\}_{m=1}^{M_{BS}}$ follow a global data distribution \mathcal{P} . In each epoch of meta training stage, the model is trained with a small amount of data from multiple local sub-tasks. Note the training data in each epoch constitute a local data distribution p_t , where $t \in \{1, 2, \dots, T\}$, T is the total number of iterations. The

Algorithm 2 meta learning algorithm for beam prediction

Input: Source tasks: $\{\mathcal{T}_m^S\}_{m=1}^{M_{BS}}$, Target tasks: $\{\mathcal{T}_n^T\}_{n=1}^{N_{BS}}$, inner-task learning rate: α_g , across-task learning rate: γ , number of update steps for inner-task training: G_m , number of update steps for adaption: G_a , number of epochs for out loop: T .

Output: Encoder part parameter: θ_e , decoder part parameter: θ_d , predicted beam based on $\{\mathcal{D}_{Qry}(n)\}_{n=1}^{N_{BS}}$.

- 1: **Meta Training**
- 2: Randomly initialize θ_d
- 3: **for** $t = 1, \dots, T$ **do**
- 4: Randomly select a task \mathcal{T}_m from $\{\mathcal{T}_m^S\}_{m=1}^{M_{BS}}$
- 5: Generate corresponding datasets \mathcal{D}_{Spt}^S and \mathcal{D}_{Qry}^S
- 6: **for** $g = 0, \dots, G_m$ **do**
- 7: obtain feature embedding: $\mathbf{E}_{Spt} = f_{enc}(\mathcal{D}_{Spt}^S, \theta_e^t)$
- 8: compute prototypical feature averaging using Eq. (13)
- 9: update α_g and θ_e^t using Eq. (14)
- 10: **end for**
- 11: obtain feature embedding: $\mathbf{E}_{Qry} = f_{enc}(\mathcal{D}_{Qry}^S, \theta_e^t)$
- 12: update θ_e^t and θ_d^t using Eq.(15)
- 13: **end for**
- 14: **Meta Adaption and Testing**
- 15: **for** $n = 1, \dots, N_{BS}$ **do**
- 16: generate the dataset $\mathcal{D}_{Spt}^T(n)$ and $\mathcal{D}_{Qry}^T(n)$ from $\{\mathcal{T}_n^T\}_{n=1}^{N_{BS}}$
- 17: load parameter of encoder part θ_e and decoder part θ_d
- 18: **for** $g = 0, \dots, G_a$ **do**
- 19: obtain feature embedding: $\mathbf{E}_{Spt} = f_{enc}(\mathcal{D}_{Spt}^T, \theta_e)$
- 20: update θ_d using Eq. (16)
- 21: **end for**
- 22: obtain feature embedding: $\mathbf{E}_{Tst} = f_{enc}(\mathcal{D}_{Qry}^T, \theta_e)$
- 23: predict the beam \mathbf{f} on \mathbf{E}_{Tst} with $f_{dec}(\theta_d)$
- 24: **end for**

mean and variance of the local data distribution $\{d_t\}_1^T$ are obviously biased compared to that of the global data distribution \mathcal{P} . Follow the analysis in the [11], let ϕ denotes the network initialization parameters, and \mathbf{W}_m denotes the set of optimal parameters for task \mathcal{T}_m . The purpose of model-agnostic meta-learning is to find ϕ that the distance $dis(\phi, \mathbf{W}_m)$ is small for all tasks,

$$\underset{\phi}{\text{minimize}} \mathbb{E}\left[\frac{1}{2} \sum_{m=1}^{M_{BS}} dis(\phi, \mathbf{W}_m)^2\right]. \quad (17)$$

Given a non-pathological set $\mathbb{S} \subset \mathbb{R}^d$, then for all points in $\phi \subset \mathbb{R}^d$, the gradient of the squared distance $dis(\phi, \mathbb{S})$ is $2(\phi - P_{\mathbb{S}}(\phi))$, where $P_{\mathbb{S}}(\phi)$ is the projection (closest point) of ϕ onto \mathbb{S} .

$$\nabla_{\phi} \mathbb{E}\left[\frac{1}{2} \sum_{m=1}^{M_{BS}} dis(\phi, \mathbf{W}_m)^2\right] = \mathbb{E}[M_{BS}\phi - \sum_{m=1}^{M_{BS}} P_{\mathbf{W}_m}(\phi)], \quad (18)$$

where $P_{\mathbf{W}_m}(\phi) = \text{argmin}_{p \in \mathbf{W}_m} dis(p, \phi)$.

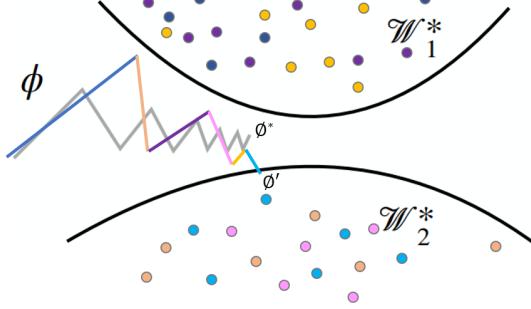


Fig. 3: Schematic diagram for the sequences of iterates obtained by moving towards two optimal solution manifolds \mathcal{W}_1^* and \mathcal{W}_2^* and converging to the points that one trends to a specific task and the other trends to both tasks respectively.

Fig. 3 shows two sequences of iterates obtained by moving alternately towards two optimal solution manifolds \mathcal{W}_1^* and \mathcal{W}_2^* and converging to the points that minimizes the average squared distance. Dots of various colors distributed in two manifolds represent training data points. We assume that dots of the same color will be sampled as a batch in one iteration. ϕ is the projection of the model initial parameters on the data space. Random few-shot sampling and stochastic gradient-based updating may guides the ϕ along the colored trajectory to the direction of an optimal ϕ' on the \mathcal{W}_2^* . Models initialized with ϕ' perform well when faced with new tasks which data are distributed nearly to \mathcal{W}_2 . However, there will be significant performance degradation in the face of new tasks which data are distributed nearly to \mathcal{W}_1 . Our proposed algorithm try to guide the ϕ to the point ϕ^* near all solution manifolds, which update trajectory is robust to stochastic effect. The gray trajectory in Fig. 3 is what we expect.

Therefore, to reduce the fluctuation during the sub-tasks acrossing, we add constraints to the update of the parameters. Let θ_e^t denotes the parameter of encoder part at t -th outer loop, which is regard as the initial parameter for the following inner loop. The parameter of $f_{enc}(\theta_e)$ is updated by

$$\begin{aligned}\theta_e^{t+1} = & \theta_e^g - \alpha_g \nabla_{\theta_e^g} \sum_{n=1}^{N_s} L(f_{dec}(\mathbf{E}_{Spt}(n), \theta_e^g), \mathbf{Y}_{Spt}(n)) \\ & + \beta(\theta_e^g - \theta_e^t),\end{aligned}\quad (19)$$

where β is used to encourage the model to alleviate overfitting to one specific sub-task. Note that, in Eq.(19), training data from N_s classes is random selected to form the data batch in each outer loop. The reason for this is that the data distribution of different classes is not uniform, so as to avoid that the classes with small data rarely appear in the training. Given proper initialization and effective training, MBBP is more suitable for tackling data scarcity during model transferring. Facing a new environment, MBBP can use a small amount of data to fine-tune the network through few-shot gradient descents.

V. THEORETICAL ANALYSIS FOR THE GRADIENT OPTIMIZATION OF MBBP

MBBP can finds a group of initial parameters which is general for different environments in the parameter space. In this section, we will derive the effective objective of MBBP and demonstrate what happens to the effective gradients computed in MBBP. Follow the conventions from Nichol & Schulman in [23], we also assume that each task gives us a sequence of k loss functions L_1, L_2, \dots, L_k , this allows us to extrapolate an effective gradient that is a function of the number of steps taken. Before we begin, let us define the following terms from [23].

$$g_i = L'_i(\phi_i) \quad (\text{gradient obtained during SGD}). \quad (20)$$

$$\phi_{i+1} = \phi_i - \alpha g_i \quad (\text{update of parameter}). \quad (21)$$

$$\bar{g}_i = L'_i(\phi_1) \quad (\text{gradient at initial point}). \quad (22)$$

$$\bar{H}_i = L''_i(\phi_1) \quad (\text{Hessian at initial point}). \quad (23)$$

For each of these definitions, $i \in [1, k]$. First, let's calculate the normal SGD gradients to $O(a^2)$ as follows,

$$\begin{aligned}g_i &= L'_i(\phi_1) + L''_i(\phi_i - \phi_1) + O(\alpha^2) \quad (\text{Taylor's theorem}) \\ &= \bar{g}_i + \bar{H}_i(\phi_i - \phi_1) + O(\alpha^2) \\ &= \bar{g}_i - \alpha \bar{H}_i \sum_{j=1}^{i-1} g_j + O(\alpha^2) \quad (\text{using } \phi_i - \phi_1 = -\alpha \sum_{j=1}^{i-1} g_j) \\ &= \bar{g}_i - \alpha \bar{H}_i \sum_{j=1}^{i-1} \bar{g}_j + O(\alpha^2).\end{aligned}\quad (24)$$

For simplicity, we define U_i as the operator that updates the parameter vector on minibatch i : $U_i(\phi) = \phi - \alpha L_i(\phi)$. The gradient of MAML can be expand as

$$\begin{aligned}g_{MAML} &= \frac{\partial}{\partial \phi_1} L_k(U_{k-1}(U_{k-2}(\dots(U_1(\phi_1)))))) \\ &= U'_1(\phi_1) \cdots U'_{k-1}(\phi_{k-1}) L'_k(\phi_k) \\ &= (I - \alpha L'_1(\phi_1)) \cdots (I - \alpha L'_{k-1}(\phi_{k-1})) L'_k(\phi_k) \\ &= \left(\prod_{j=1}^{k-1} (I - \alpha L''_j(\phi_j)) \right) g_k \\ &= \left(\prod_{j=1}^{k-1} (I - \alpha \bar{H}_j) \right) \left(\bar{g}_k - \alpha \bar{H}_k \sum_{j=1}^{k-1} \bar{g}_j \right) + O(\alpha^2) \\ &= \bar{g}_k - \alpha \sum_{j=1}^{k-1} \bar{H}_j \bar{g}_k - \alpha \bar{H}_k \sum_{j=1}^{k-1} \bar{g}_j + O(\alpha^2).\end{aligned}\quad (25)$$

Next, let's analysis the gradient update in MBBP inner loop. For each update step, the weight is calculated as follow,

$$\theta_{i+1} = \theta_i - \alpha g_i + \beta(\theta_i - \theta_{i-1}). \quad (26)$$

Thus, the weight difference item $\phi_i - \phi_1$ can be expand as

$$\begin{aligned}\phi_i - \phi_1 &= -\alpha \frac{1 - \beta^{i-1}}{1 - \beta} g_1 - \alpha \frac{1 - \beta^{i-2}}{1 - \beta} g_2 \cdots - \alpha \frac{1 - \beta}{1 - \beta} g_{i-1} \\ &= -\alpha \sum_{j=1}^{i-1} \frac{1 - \beta^j}{1 - \beta} g_{i-j}.\end{aligned}\quad (27)$$

Then, substitute eq.27 into eq.24, we can derive an expression for the novel gradient g^{MBBP} for each training step

$$g_i^{MBBP} = \bar{g}_i - \alpha \bar{H}_i \sum_{j=1}^{i-1} \frac{1-\beta^j}{1-\beta} g_{i-j} + O(\alpha^2). \quad (28)$$

Remark 1: Through k steps of iteration, the gradient g_{MBBP} in our algorithm can thus be expressed as

$$\begin{aligned} g_{MBBP} &= \left(\prod_{j=1}^{k-1} (I - \alpha \bar{H}_j) \right) g_k^{MBBP} \\ &= \bar{g}_k - \alpha \bar{H}_k \sum_{j=1}^{k-1} \frac{1-\beta^j}{1-\beta} \bar{g}_{k-j} - \alpha \sum_{j=1}^{k-1} \bar{H}_j \bar{g}_k + O(\alpha^2). \end{aligned} \quad (29)$$

Note that the main difference between g_{MAML} and g_{MBBP} lies in the coefficient of the second item in the eq.29, which reflects the gradient inner product between different iteration steps.

For simplicity of exposition, let's consider the $k = 4$ case,

$$\begin{aligned} g_{MAML} &= \bar{g}_4 - \alpha \bar{H}_1 \bar{g}_4 - \alpha \bar{H}_2 \bar{g}_4 - \alpha \bar{H}_3 \bar{g}_4 \\ &\quad - \alpha \bar{H}_4 \bar{g}_1 - \alpha \bar{H}_4 \bar{g}_2 - \alpha \bar{H}_4 \bar{g}_3 + O(\alpha^2). \end{aligned} \quad (30)$$

$$\begin{aligned} g_{MBBP} &= \bar{g}_4 - \alpha \bar{H}_4 \bar{g}_3 - \alpha \bar{H}_3 \bar{g}_4 - \alpha(1+\beta) \bar{H}_4 \bar{g}_2 - \alpha \bar{H}_2 \bar{g}_4 \\ &\quad - \alpha(1+\beta+\beta^2) \bar{H}_4 \bar{g}_1 - \alpha \bar{H}_1 \bar{g}_4 + O(\alpha^2). \end{aligned} \quad (31)$$

Then we follow Nichol & Schulman [23] and define two items: G_{avg} and G_{inr} and show the interesting expressions of g_{MAML} and g_{MBBP} . G_{avg} is defined as gradient of expected loss,

$$G_{avg} = \mathbb{E}_{\tau,1}[\bar{g}_1]. \quad (32)$$

G_{avg} is the gradient direction of “joint training” problem, which is the mean loss over tasks. G_{inr} is defined as follows

$$\begin{aligned} G_{inr} &= \mathbb{E}_{\tau,i,j}[\bar{H}_j \bar{g}_i] = \mathbb{E}_{\tau,i,j}[\bar{H}_i \bar{g}_j] \\ &= \frac{1}{2} \mathbb{E}_{\tau,i,j}[\bar{H}_i \bar{g}_j + \bar{H}_j \bar{g}_i] \\ &= \frac{1}{2} \mathbb{E}_{\tau,i,j} \left[\frac{\partial}{\partial \phi_i} (\bar{g}_i \cdot \bar{g}_j) \right]. \end{aligned} \quad (33)$$

$G_{inr}(k)$, $k = |i - j|$, is the direction that increases the inner product between gradients of i -th and j -th minibatches for a given task. Now, recalling the gradient expression, we get the following expressions for the meta-gradients, for $k = 4$,

$$\begin{aligned} \mathbb{E}[g_{MAML}] &= (1)G_{avg} - (2\alpha)G_{inr}(1) - (2\alpha)G_{inr}(2) \\ &\quad - (2\alpha)G_{inr}(3) + O(\alpha^2). \end{aligned} \quad (34)$$

$$\begin{aligned} \mathbb{E}[g_{MBBP}] &= (1)G_{avg} - (2\alpha)G_{inr}(1) - (2\alpha(1+\beta))G_{inr}(2) \\ &\quad - (2\alpha(1+\beta+\beta^2))G_{inr}(3) + O(\alpha^2). \end{aligned} \quad (35)$$

In practice, the G_{avg} term ensures the model converge towards the minimum of the expected loss over all tasks. The coefficient of the G_{inr} item increases nonlinearly with k , which enhances the inner product of gradients between different tasks. For example, if the gradients’ directions of two tasks are unidirectional, the negative inner product in MBBP will be further smaller than that in MAML due to

the β . Conversely, if the gradients’ directions of two tasks are opposite, the negative inner product in MBBP will be further larger than that in MAML.

Remark 2: The G_{inr} item will guides the model concern on reducing the expected loss of tasks with large differences. Compared with MAML, MBBP adjusts the coefficient of the inner product between different epochs non-linearly though the β . Since β is between 0 and 1, MBBP’s attention to the gradient inner product grows gradually but not explosively, which avoids being stuck into local minimums and meta-overfitting during the training process.

VI. SIMULATION RESULTS

In this section, we evaluate the performance of MBBP using numerical simulations. We first introduce the simulation scenarios, comparison algorithms, and metrics used in the experiments. Secondly, we evaluate the robustness to SNR and impact of mmWave antennas array size for MBBP and other algorithms on a public dataset DeepMIMO. Then we utilize the t-SNE to visualize the performance of MBBP and compare it with the direct transfer learning algorithm. Finally, we measure the impact of changes in the amount of data and adaptation gradstep on the performance of MBBP in the process of model transfer.

A. Simulation Scenarios

We utilize the scenario ‘O1’ in the publicly available dataset DeepMIMO [38] for simulations. There are two generator scripts in the ‘O1’ scenario used to generate two channel sets at sub-6GHz and mmWave with 3D ray-tracing. In this scenario, there are 18 BSs, numbered from 1 to 18, located along two vertical streets with plenty of users scattered on. We adopt 16 BSs numbered from 2 to 17, and sample users in a grid in front of each BS. Each BS is equipped with two co-located uniform linear arrays (ULAs) at 28GHz and 3.5GHz. The spacing between the elements of the BS antenna array is half-wavelength. The bandwidth is 0.5 GHz at the mmWave channel and 0.02 GHz at the sub-6GHz channel. The number of orthogonal frequency-division multiplexing (OFDM) subcarriers is 512 and 32 at mmWave and sub-6GHz channel, respectively. The subcarriers are ordered according to their received power. We calculate the channel with the first 5 and 15 subcarriers. The users’ uplink channels and optimal downlink beams are collected as the sample pairs for the corresponding BSs. The detail parameters for DeepMIMO dataset generation are shown in Table I. The dataset is generated as described in IV-B.

B. Model Prediction Performance Evaluation

In this section, we utilize the datasets introduced in section VI-A to evaluate the performance of MBBP. We will also evaluate the impact of various SNR and the number of antennas on the model prediction performance. In order to compare objectively and avoid the influence of artificial sampling, the basic net is trained with one BS selected randomly from all BSs. The learning rate in the basic training drops ten times

TABLE I: DeepMIMO Dataset Parameters

| Parameter | Value | |
|----------------------------|----------|--------|
| Scenario name | O1_28 | O1_3p5 |
| Active BS | 2-17 | 2-17 |
| Number of BS antennas | 16,32,64 | 4 |
| Antenna spacing | 0.5 | 0.5 |
| Bandwidth(GHz) | 0.5 | 0.02 |
| Number of OFDM subcarriers | 512 | 32 |
| OFDM sampling factor | 1 | 1 |
| OFDM limit | 32 | 32 |
| Number of paths | 5 | 15 |

every 10,000 epochs. The hyper-parameters of basic training are summarized in Table II. Then, in the meta-learning stage, we randomly select multiple BSs for meta training and the rest BSs for testing. The ratio of BSs used for learning to BSs used for testing is 5:3. Note that BS selected either in the basic training stage or the meta-training stage will not appear in the testing. The hyper-parameters of MBBP are summarized in Table III.

TABLE II: Basic Training Hyper-Parameters

| Parameter | Value |
|------------------------|---------------------|
| Solver | ADAM |
| Learning rate | 1e-3 |
| Learning rate schedule | 0.1 @ 10000th epoch |
| Dropout rate | 0.5 |
| Total epochs | 30000 |

1) *Comparison Evaluation Algorithms*: We compare MBBP with three different algorithms. (1) Direct learning (DL): training a network for the target BS from scratch with sufficient or limited data. (2) Direct transfer learning (DTL): training a network on source BSs and finetuning the network on target BS with limited data. (3) Classic meta-learning algorithm: MAML. For fairly comparing the performance, we adopt same network for each algorithms and some implementation details have been adjusted individually. For direct learning, regardless of whether the data is sufficient or limited, we will train the network until the loss no longer changes. For direct transfer learning, we train the network with data collected in the source BSs, and fine-tuning to other BSs with an equal number of adaption samples used in MBBP. For MAML, we use its first-order approximation algorithm for memory efficiency and train in the same way as MBBP. All experiments are done on a machine with a TESLA V100 GPU.

2) *Performance Evaluation Metrics*: Three metrics are used for performance evaluation, including: (1) Top-1 accuracy, which is defined as

$$Accuracy = \frac{\sum_{d=1}^{D_t} \text{bool}(\mathbf{f}_d^* = \mathbf{f}'_d)}{D_t}, \quad (36)$$

where D_t denotes the number of data in the test dataset, and $\text{bool}(\cdot)$ denotes the boolean function where output is 1 if $\mathbf{f}_d^* = \mathbf{f}'_d$ otherwise output is 0. Accuracy reflects the capability of the trained model whether to select the optimal beam. Note that in the case of a large-scale codebook, UE may obtain very close achievable rates by using two adjacent beams,

TABLE III: Meta Training And Adaption Hyper-Parameters

| Parameter | Value |
|---|------------------------|
| Inner-task initial learning rate α | 1e-3 |
| Across-task learning γ | 1e-4 |
| Momentum β | 0.9 |
| Task split (source : target) | 5:3 |
| Number of sub-tasks in each task | 16, 32, 64 |
| Number of samples in support dataset | 10 |
| Number of samples in query dataset | 20 |
| Number of samples in adaption dataset | 10 |
| Number of samples in test dataset | 100 |
| Dataset size for each task | $\approx 1.1\text{e}3$ |
| Number of gradstep G_m | 50 |

so this indicator may not accurately reflect the performance. However, in the case of a small-scale codebook, this indicator can accurately reflect the discriminant ability of the model. (2) Top-3 accuracy, which is also adopted in [19]. This indicator measures whether the optimal beam is within the top-3 predictions instead of only focusing on the Top-1 prediction. (3) Mean spectral efficiency (MSE), which is defined as

$$\text{MSE} = \frac{\sum_{d=1}^{D_t} \frac{R_d}{B}}{|D_t|} (\text{bit/s/Hz}), \quad (37)$$

where R_d is the rate achieved at d -th data point, B is the system bandwidth. MSE can directly reflects the reliability performance of the beam predicted by the model.

3) *impact of various SNR*: To evaluate the robustness for noise interference of MBBP and other methods, we first deploy an antenna array with 32 elements to the BSs, then conduct a few experiments in a different SNR regimes. Note that this SNR refers to the sub-6GHz and mmWave receive SNR, i.e., the radio of the signal to noise in the channel measurement.

TABLE IV: Top-3 Accuracy For Beam Prediction

| SNR(dB) | -5 | 0 | 5 | 10 | 15 | 20 | 25 | 30 |
|---------|------|------|------|------|------|------|------|------|
| DL* | 0.76 | 0.91 | 0.93 | 0.96 | 0.97 | 0.97 | 0.98 | 0.98 |
| MBBP | 0.74 | 0.83 | 0.89 | 0.93 | 0.94 | 0.95 | 0.95 | 0.95 |
| MAML | 0.62 | 0.65 | 0.83 | 0.89 | 0.89 | 0.89 | 0.91 | 0.91 |
| DTL | 0.43 | 0.58 | 0.64 | 0.66 | 0.68 | 0.69 | 0.69 | 0.69 |
| DL | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.07 |

In Fig. 4(a), we evaluate Top-1 accuracy of the mmWave beam prediction for different SNR regimes. When the overheads of data collection is not taken into account, i.e., sufficient adapting data are available, the direct learning method can be regarded as the theoretical upper bound of learning-based approaches, which reach 80% Top-1 accuracy when the SNR is above 20dB. However, when adapting data is limited, two meta-learning-based methods (MAML and MBBP) obviously outperform the direct transfer learning and direct learning methods. The accuracy of the direct learning method is almost zero, meaning that the model trained with limited data is useless for unseen new tasks. The main reason is that the traditional deep learning model cannot converge with limited training data. The performance of MBBP is better than MAML and DTL in different noise situations and the closest to the upper bound. In addition, with the increase of SNR, the improvement of accuracy of each algorithm gradually

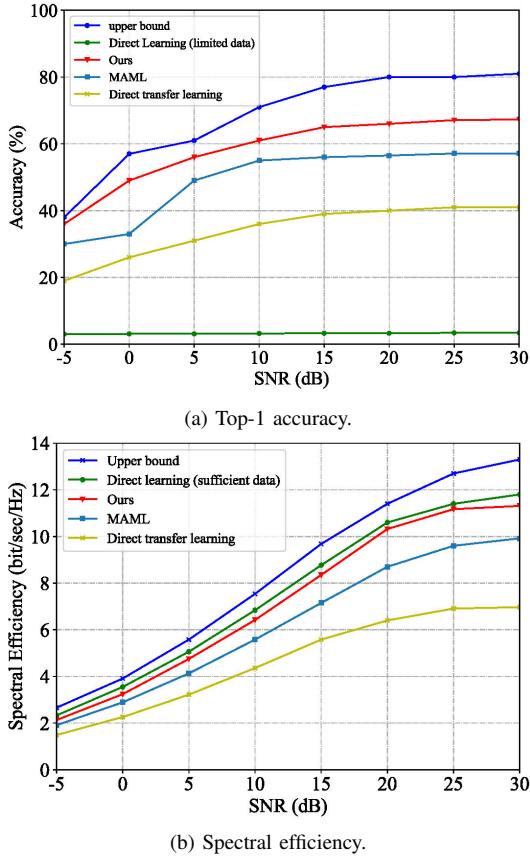


Fig. 4: Performance evaluation under different SNR.

decreases. When SNR is greater than 15dB, increasing SNR has little effect on accuracy.

In Fig. 4(b), we evaluate the performance using MSE. The upper bound (dark blue line) here is obtained by the optimal beam for each data in the test dataset. It is apparent that the MSE of MBBP is better than DTL and MAML, and very close to the DL, which trained with sufficient data can be regarded as the upper bound of learning-based approaches. Compared with the upper bound, MBBP has only 10.9% degradation at most, and the DL with sufficient data has 10.89%. With the overheads of data collection factored in, MBBP is advantageous compared to DL as evidenced by the MSE results. In addition, compared with the exhaustive search method, MBBP has more advantages in the overheads of beam training. Interestingly, even the accuracy is lower than the direct learning method, the difference in MSE is almost negligible. This phenomenon corroborates the point we mentioned in section VI-B(2), the difference in the achievable rate obtained by some UEs through two adjacent beams is negligible, learning-based methods even not select the optimal beam stably but still pick the beam closest to the optimal one.

In table IV, we evaluate the Top-3 accuracy of the mmWave beam prediction for different SNR regimes. Note that the DL* in the table represents the DL algorithm trained with sufficient data. And DL in the table represents the DL algorithm trained with limited data. The simulation result shows that MBBP outperforms other algorithms and is closest to the DL*, while

the DL method can hardly predict the optimal beam. It proves that, when MBBP model is deployed in a new environment, it can also offer reasonable candidate beams based on a small amount of historical data, which greatly reduces the scanning range required for initial beam establishment and alignment.

4) *impact of mmWave antennas array sizes:* Then we evaluate the impact of mmWave antennas array sizes on the model prediction performance. For the mmWave beam prediction, the increase in the number of elements in the mmWave antennas array leads to more candidate beams in the codebook, which becomes a more complex multi-classification problem.

As shown in Fig. 5, the prediction accuracy of four algorithms degrade as the size of mmWave array (in other words, the number of candidate beams) increases. Although the DL algorithm does not predict the optimal beam 100%, the spectral efficiency is very close to the upper bound (exhaustive search) no matter the antennas array sizes. It proves that the deep learning-based algorithm is one of the effective solutions to the beam prediction problem and robust to the size of the antennas array. For the DTL algorithm, its performance degrades obviously with the increase of the number of antennas, and under the same SNR condition, the maximum prediction accuracy difference between 16 beams and 64 beams reaches nearly 30%. This means that when traditional deep learning models face with complex classification problems, its full-trained model is difficult to transfer to new unseen tasks unless sufficient data is collected. For the MBBP and MAML, the performance are both better than the DTL, which means that the meta-learning-based algorithm can effectively transfer the model in the case of few shots. Note that meta-overfitting easily appears during training MAML, which affects the performance in the testing. The performance of MBBP is very close to that of DL no matter the antennas array sizes, but the former utilizes much less data than the latter (10 : 1.1e3).

C. Latent Embedding Visualization

In order to show the classification performance of MBBP intuitively, we utilize t-SNE [39] to visualize the representations of all test data points before they were put into decoder part. t-SNE is essentially an embedded model that can map data from a high-dimensional space to a low-dimensional space and retain local characteristics of data sets, which is often adopted in the image classification task. In this simulation, we employ the query dataset \mathcal{D}_{Qry} of one BS (with 32 candidate beams) at 15 dB SNR in the $\{\mathcal{T}_n\}_{n=1}^{N_{BS}}$.

Figure 6 (a) and (b) show the projection of feature for the test dataset via MBBP and DTL, respectively, in which each dot represents the feature embedding of a data point and the colors represent the labels (optimal beam index) of the data points. In Fig.6 (a), almost all of the classes are clearly separated, i.e., large inter-class distances, and points of the same category are grouped into an independent cluster, i.e., small in-class distances. This means that MBBP can learn more discriminating features to separate different categories of data. However, in Fig.6 (b), points belonging to various categories are mixed together without sharp boundary, which means that the directly transfer learning method can not

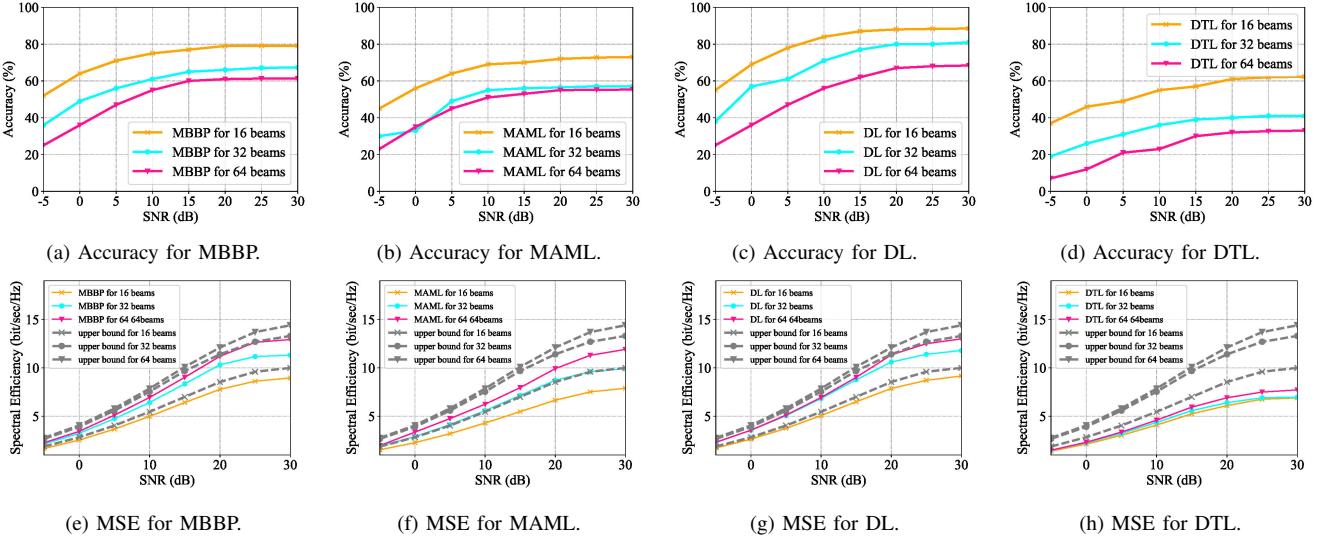


Fig. 5: Performance evaluation under different antennas array sizes.

distinguish the category of test data. When deployed on the base station side, this model may output the wrong beam to the user.

In Fig.6 (c), (d), (e), (f), we further visualized the classification performance of MBBP and DTL for two specific categories of test data through t-SNE. We highlight the data points labeled with the No.1 beam in the codebook and the No.11 beam. Users served by these beams are usually to be distributed on the sides and the middle of the sector, respectively. It can be observed that for both algorithms, the data points labeled with No.1 beam are clustered together. The data points under MBBP method are more closely distributed and there is a very obvious gap between them and other sample points. For the No.11 beam, the data points under the DTL method are obviously scattered, which means that DTL is difficult to distinguish these data from the test dataset. However, the data points under MBBP are more aggregated.

D. Assessment of Transfer Needs

To answer the questions in section III-B, the model transfer needs required by the MBBP are verified in this section. We evaluate the MSE of MBBP against the number of adapting data and the gradstep G_m for adaption. To investigate the impacts of varying adaptation samples, we conduct the simulations with 1,3,5,7 and 10 samples at 0 dB and 10 dB SNR. Note that the upper bound is the MSE of optimal beam for each data point. The results are shown in Fig. 7(a). The MSE of MBBP improves with the number of adaption samples increases. This phenomenon is more significant in a higher SNR environment, which is also consistent with the situation that the performance of the learning-based model increases as the larger the amount of the available data. However, we also found that the performance growth rate decreased gradually with the increase of adapting data. This indicates that the performance gain of the deep learning model will gradually lose its sensitivity to the amount of data. Therefore, in the practical communication environment, the performance gain

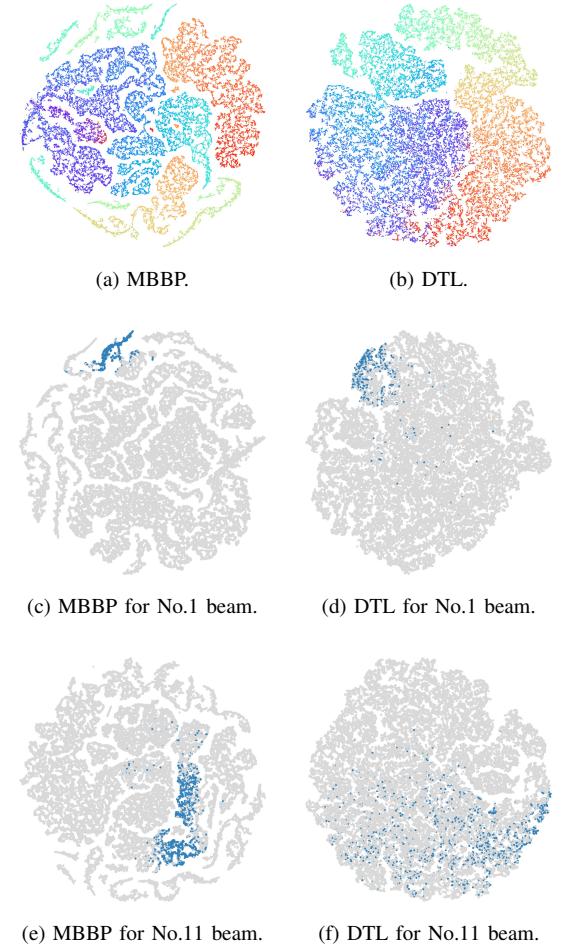
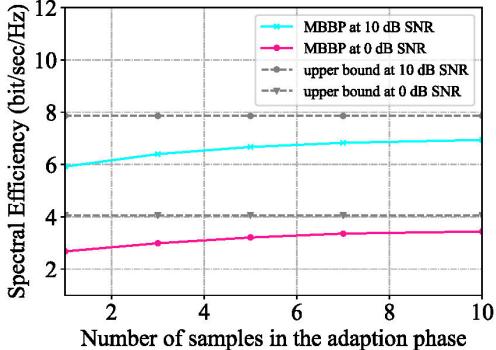
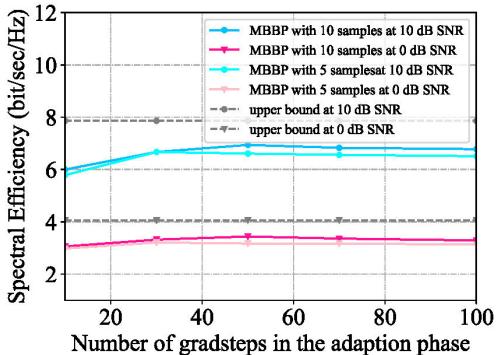


Fig. 6: Performance evaluation under different antennas array sizes.

brought by the deep learning model and the overheads of data collection will be an interesting trade-off problem.



(a) The Performance versus the number of adaptation samples.



(b) The Performance versus the number of adaptation gradstep.

Fig. 7: Performance evaluation under different numbers of adaption samples and gradstep.

To investigate the impact of varying adaptation gradstep, we conduct the simulations with 5, 10 samples at 0 dB and 10dB SNR. The results are shown in Fig. 7(b). The MSE of MBBP first improves and then drops as the gradstep increases. Note that the number of gradstep required to achieve the highest performance increases as the number of samples. During meta-training, the distribution of batch data determines the direction of model updating, and the number of gradstep determines the amplitude of model advancing along this direction. A large number of gradstep may lead to the model falling into a local optimum, otherwise may lead to the model underfitting. In addition, the computation overheads of large gradstep is also high. Therefore, the number of gradstep should be selected according to the number of samples empirically.

VII. CONCLUSION

In this paper, we formulated the millimeter wave (mmWave) beam prediction using sub-6GHz channels state information as a multiclass-classification problem and utilized the deep learning method to give a solution. Considering the challenges of model transferring in the wireless communication system, we proposed a model-agnostic beam prediction algorithm based on meta-learning. We evaluated our algorithm on the publicly available dataset DeepMIMO, then compared it with

direct learning, direct transfer learning, and MAML. Simulation results showed that MBBP could effectively transfer the model to a new environment with few data. Under different SNR levels, MBBP could achieve high prediction accuracy and spectral efficiency, proving its superiority and robustness. We also evaluated the transfer overheads of MBBP. The results showed that MBBP could obtain good performance with a small cost of data collection and few iterative training.

REFERENCES

- [1] C. N. Barati, S. A. Hosseini, S. Rangan, P. Liu, T. Korakis, S. S. Panwar, and T. S. Rappaport, "Directional cell discovery in millimeter wave cellular networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 12, pp. 6664–6678, 2015.
- [2] J. Choi, "Beam selection in mm-Wave multiuser MIMO systems using compressive sensing," *IEEE Transactions on Communications*, vol. 63, no. 8, pp. 2936–2947, 2015.
- [3] Y. Wang, A. Klautau, M. Ribero, M. Narasimha, and R. W. Heath, "Mmwave vehicular beam training with situational awareness by machine learning," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.
- [4] J. Wang, L. Zhou, C. W. Pyo, T. Baykas, C. S. Sum, M. A. Rahman, G. Jing, R. Funada, F. Kojima, and H. Harada, "Beam codebook based beamforming protocol for multi-Gbps millimeter-wave WPAN systems," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, pp. 1390–1399, 2009.
- [5] J. Wang, Z. Lan, C. S. Sum, C. W. Pyo, and S. Kato, "Beamforming codebook design and performance evaluation for 60GHz Wideband WPANs," in *Vehicular Technology Conference Fall (VTC 2009-Fall)*, 2009 IEEE 70th, 2009.
- [6] O. Semari, W. Saad, M. Bennis, and M. Debbah, "Integrated millimeter wave and Sub-6 GHz wireless networks: A roadmap for joint mobile broadband and ultra-reliable low-latency communications," *IEEE Wireless Communications*, 2018.
- [7] A. Ali, N. González-Prelcic, and J. Heath, "Millimeter wave beam-selection using out-of-band spatial information," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, 2017.
- [8] B. Coll-Perales, J. Gozalvez, and M. Gruteser, "Sub-6 GHz assisted mac for millimeter wave vehicular communications," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 125–131, 2019.
- [9] M. Hashemi, C. E. Koksal, and N. B. Shroff, "Dual Sub-6 GHz – millimeter wave beamforming and communications to achieve low latency and high energy efficiency in 5G systems," *IEEE Transactions on Communications*, pp. 1–1, 2017.
- [10] H. Ye, G. Y. Li, and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communication Letters*, vol. PP, no. 99, pp. 114–117, 2017.
- [11] X. Wei, C. Hu, and L. Dai, "Deep learning for beamspace channel estimation in millimeter-wave massive MIMO systems," *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 182–193, 2021.
- [12] A. Klautau, N. Gonzalez-Prelcic, and R. Heath, "LIDAR data for deep learning-based mmWave beam-selection," *Wireless Communications Letters, IEEE*, vol. 8, no. 3, pp. 909–912, 2019.
- [13] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari, and F. Adachi, "Deep learning for physical-layer 5G wireless techniques: Opportunities, challenges and solutions," *IEEE Wireless Communications*, 2019.
- [14] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep reinforcement learning for 5G networks: Joint beamforming, power control, and interference coordination," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1581–1592, 2020.
- [15] Y. Zhang, Y. Mu, Y. Liu, T. Zhang, and Y. Qian, "Deep learning-based beamspace channel estimation in mmWave massive MIMO systems," *IEEE Wireless Communication Letters*, vol. PP, no. 99, pp. 1–1, 2020.
- [16] A. Ahmed, A. Sam, V. Paul, L. Ying, Q. Qu, and T. Djordje, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. PP, pp. 1–1, 2018.
- [17] S. S. Min, Y. G. Lim, H. P. Sang, L. Dai, and C. B. Chae, "Deep learning-based mmWave beam selection for 5G NR/6G with Sub-6 GHz channel information: Algorithms and prototype validation," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2020.
- [18] K. Ma, P. Zhao, and Z. Wang, "Deep learning assisted beam prediction using out-of-band information," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020.

- [19] M. Alrabeiah and A. Alkhateeb, "Deep learning for mmWave beam and blockage prediction using Sub-6GHz channels," *IEEE Transactions on Communications*, vol. PP, no. 99, pp. 1–1, 2020.
- [20] Fei-Fei, Li, Fergus, Rob, Perona, and Pietro, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2006.
- [21] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *ICLR*, 2017.
- [22] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017.
- [23] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *ArXiv*, vol. abs/1803.02999, 2018.
- [24] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," *ArXiv*, vol. abs/1807.05960, 2019.
- [25] O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *NIPS*, 2016.
- [26] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *NIPS*, 2017.
- [27] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.
- [28] T. Mensink, J. J. Verbeek, F. Perronnin, and G. Csurka, "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost," in *ECCV*, 2012.
- [29] M. Andrychowicz, M. Denil, S. G. Colmenarejo, M. W. Hoffman, D. Pfau, T. Schaul, and N. de Freitas, "Learning to learn by gradient descent by gradient descent," in *NIPS*, 2016.
- [30] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations*, 2017. [Online]. Available: <https://openreview.net/forum?id=rJY0-Kcll>
- [31] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.
- [32] H. Qi, M. A. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5822–5830, 2018.
- [33] S. Park, O. Simeone, and J. Kang, "End-to-end fast training of communication links without a channel model via online meta-learning," *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, 2020.
- [34] Y. Yang, F. Gao, Z. Zhong, B. Ai, and A. Alkhateeb, "Deep transfer learning-based downlink channel prediction for FDD massive MIMO systems," *IEEE Transactions on Communications*, vol. 68, pp. 7485–7497, 2020.
- [35] H. Mao, H. Lu, Y. Lu, and D. Zhu, "Roemnet: Robust meta learning based channel estimation in OFDM systems," *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2019.
- [36] K. Thar, T. Z. Oo, Z. Han, and C. S. Hong, "Meta-learning-based deep learning model deployment scheme for edge caching," *2019 15th International Conference on Network and Service Management (CNSM)*, pp. 1–6, 2019.
- [37] H.-H. Wu, Z. Zhang, C. Jiao, C. Li, and T. Q. S. Quek, "Learn to sense: A meta-learning-based sensing and fusion framework for wireless sensor networks," *IEEE Internet of Things Journal*, vol. 6, pp. 8215–8227, 2019.
- [38] A. Alkhateeb, "Deepmimo: A generic deep learning dataset for millimeter wave and massive mimo applications," *ArXiv*, vol. abs/1902.06435, 2019.
- [39] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE." *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [40] R. W. Heath, N. González-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, 2016.
- [41] P. Zhou, X. Fang, X. Wang, Y. Long, R. He, and X. Han, "Deep learning-based beam management and interference coordination in dense mmWave networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 592–603, 2018.
- [42] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4068–4076.
- [43] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, "Rapid learning or feature reuse? towards understanding the effectiveness of MAML," *CoRR*, vol. abs/1909.09157, 2019. [Online]. Available: <http://arxiv.org/abs/1909.09157>
- [44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.



Ruming Yang received the bachelor's degree and master's degree in Electronic and electrical engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2016 and 2019, where he is currently pursuing the Ph.D. degree in information and communication engineering with the School of Information Science and Engineering. His research interests mainly focus on intelligent wireless communications.



Zhengming Zhang received the B.S. degree in electronic information science and technology from Nanjing Agricultural University, Nanjing, China, in 2016. He is currently pursuing the Ph.D. degree in information and communication engineering with the School of Information Science and Engineering, Southeast University. His current research interests include wireless big data, distributed machine learning, 5G mobile networks, UAV aided communication, and resource management.



Xiangyu Zhang received the B.S. degree in the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China, in 2015. He is currently pursuing the Ph.D. degree in information and communication engineering with the School of Information Science and Engineering, Southeast University. His current research interests include Intelligence wireless resource management, 5G mobile networks, Near-field communication, UAV aided communication, and Model-based learning.



Chunguo Li Chunguo Li (SM'16) received the bachelor's degree in wireless communications from Shandong University in 2005, and Ph.D. degree in wireless communications from Southeast University in 2010. In July 2010, he joined the Faculty of Southeast University, Nanjing China, where he was Associate Professor between 2012 and 2016, and Full Professor since 2017 to present. From June 2012 to June 2013, he was the Postdoctor with Concordia University, Montreal, Canada. From July 2013 to August 2014, he was with the DSL laboratory of Stanford University as Visiting Associate Professor. From August 2017 to July 2019, he was the adjunct professor of Xizang Minzu University under the supporting Tibet program organized by China National Human Resources Ministry.

He is the Fellow of IET, Fellow of China Institute of Communications (CIC), Chair of IEEE Computational Intelligence Society Nanjing Chapter, and Chair of Advisory Committee for Instruments industry in Jiangsu province. He has served as editor for a couple of international journals and as session chair for many international conferences. His research interests are in 6G cell-free distributed MIMO wireless communications, and AI based image signal processing.



Yongming Huang (M'10-SM'16) received the B.S. and M.S. degrees from Nanjing University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from Southeast University, Nanjing, in 2007. Since March 2007 he has been a faculty in the School of Information Science and Engineering, Southeast University, China, where he is currently a full professor. He has also been the Director of the Pervasive Communication Research Center, Purple Mountain Laboratories, since 2019.

From 2008 to 2009, he visited the Signal Processing Lab, Royal Institute of Technology (KTH), Stockholm, Sweden. He has published over 200 peer-reviewed papers, hold over 80 invention patents. His current research interests include intelligent 5G/6G mobile communications and millimeter wave wireless communications. He submitted around 20 technical contributions to IEEE standards, and was awarded a certificate of appreciation for outstanding contribution to the development of IEEE standard 802.11aj. He served as an Associate Editor for the IEEE Transactions on Signal Processing and a Guest Editor for the IEEE Journal Selected Areas in Communications. He is currently an Editor-at-Large for the IEEE Open Journal of the Communications Society and an Associate Editor for the IEEE Wireless Communications Letters.



Luxi Yang (M'96-SM'17) received the M.S. and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 1990 and 1993, respectively. Since 1993, he has been with the Department of Radio Engineering, Southeast University, where he is currently a Full Professor of Information Systems and Communications, and the Director of the Digital Signal Processing Division. He has authored or coauthored of two published books and more than 200 journal papers, and holds 50 patents. His current research interests include

signal processing for wireless communications, MIMO communications, intelligent wireless communications, and statistical signal processing. Prof. Yang received the first and second class prizes of science and technology progress awards of the State Education Ministry of China in 1998, 2002, and 2014. He is currently a member of Signal Processing Committee of the Chinese Institute of Electronics.