Downlink CSI Feedback Algorithm with Deep Transfer Learning for FDD Massive MIMO Systems

Jun Zeng, *Graduate Student Member, IEEE*, Jinlong Sun, *Member, IEEE*, Guan Gui, *Senior Member, IEEE*, Bamidele Adebisi, *Senior Member, IEEE*, Tomoaki Ohtsuki, *Senior Member, IEEE*, Haris Gacanin, *Fellow, IEEE*, and Hikmet Sari, *Life Fellow, IEEE*

***Abstract*—In this paper, a channel state information (CSI) feedback method is proposed based on deep transfer learning (DTL). The proposed method addresses the problem of high training cost of downlink CSI feedback network in frequency division duplexing (FDD) massive multiple-input multiple-output (MIMO) systems. In particular, we obtain the models of different wireless channel environments at low training cost by fine-tuning the pre-trained model with a relatively small number of samples. In addition, the effects of different layers on training cost and model performance are discussed. Furthermore, a model-agnostic meta-learning (MAML)-based method is proposed to solve the problem associated with large number of samples of a wireless channel environment required to train a deep neural network (DNN) as a pre-trained model. Our results show that the performance of the DTL-based method is comparable with that of the DNN trained with a large number of samples, which demonstrates the effectiveness and superiority of the proposed method. At the same time, although there is a certain performance loss compared with the DTL-based method, the MAML-based method shows good performance in terms of the normalized mean square error (NMSE).**

***Index Terms*—Deep transfer learning (DTL), downlink CSI, limited feedback, FDD, massive MIMO, model-agnostic meta-learning (MAML).**

I. Introduction

In a frequency division duplexing (FDD) communication system, the acquisition of downlink channel state information (CSI) plays an important role in precoding, beamforming, and power allocation at the base station (BS) [1]–[4]. To reduce the feedback overhead of downlink CSI, algorithms such as compressed sensing are considered by using the partial reciprocity of uplink and downlink channels and the correlation of CSI in time and frequency domains [5]–[9].

In recent years, deep learning has attracted attention in wireless communications [10]–[25]. Hence, deep learning-based downlink CSI feedback methods have also been proposed [26]–[31]. Compared with the traditional compressed sensing algorithms, deep neural network (DNN) can achieve good reconstruction performance of downlink CSI and reduce feedback overhead. In [26], CsiNet is proposed for downlink CSI feedback. To learn the correlation between time slots in time-varying channels, the long short-term memory (LSTM) network is added to CsiNet in [27]. In [28], downlink CSI is compressed by using the correlation between the amplitudes of uplink and downlink CSI. Instead of fully connected layer, the pooling layer is used to reduce dimension in [29], which greatly reduces the number of network parameters. The networks described above assume that the feedback is perfect, that is, the BS can receive the feedback codeword without error. To simulate a realistic wireless channel environment, various noises and time delays are added to the feedback in subsequent work [30], [31]. However, there is a problem of poor generalization when using DNN to implement downlink CSI feedback. Existing works on deep learning-based downlink CSI feedback methods all focus on a given wireless channel environment [26]–[31], and when facing a new wireless channel environment, the performance of DNN declines sharply due to the model generalization problem, hence it is necessary to train a DNN from scratch with the CSI data of the new wireless channel environment. Nevertheless, in FDD massive multiple-input multiple-output (MIMO) systems with high dimension downlink CSI, this exercise requires high volume of data, which will result in high training cost.

Transfer learning is a machine learning technique, which aims to improve the performance of target tasks using the knowledge extracted from one or more source tasks. Research on transfer learning traces back to 1995 under various names: knowledge transfer, learning to learn, multi-task learning, life-long learning, etc [32]. According to the relationship between the source domain and the target domain, the transfer learning methods can be categorized into instance-transfer, feature-representation-transfer, and parameter-transfer, etc [32]. Nowadays, with the development of deep learning, how to effectively transfer knowledge by DNN has become an important research direction. Accordingly, the deep transfer learning (DTL) combined with deep learning and transfer learning is proposed [33]. Generally, training a DNN requires large

This work was supported by the Major Project of the Ministry of Industry and Information Technology of China under Grant TC190A3WZ-2, the JSPS KAKENHI under grant JP19H02142, the National Natural Science Foundation of China under Grant 61901228, the Six Top Talents Program of Jiangsu under Grant XYDXX-010, the Program for High-Level Entrepreneurial and Innovative Team under Grant CZ002SC19001, the project of the Key Laboratory of Universal Wireless Communications (BUPT) of Ministry of Education of China under Grant KFKT-2020106. *(Corresponding authors: Guan Gui; Hikmet Sari).*

J. Zeng, J. Sun, G. Gui, and H. Sari are with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (Email: {1019010408, sunjinlong, guiguan, hikmet}@njupt.edu.cn).

Bamidele Adebisi is with the Department of Engineering, Faculty of Science and Engineering, Manchester Metropolitan University, Manchester M1 5GD, UK (Email: b.adebisi@mmu.ac.uk).

T. Ohtsuki is with the Department of Information and Computer Science, Keio University, Yokohama 223-8521, Japan (e-mail: ohtsuki@ics.keio.ac.jp).

H. Gacanin is with the Faculty of Electrical Engineering and Information Technology, RWTH Aachen University, 52062 Aachen, Germany (Email: harisg@ice.rwth-aachen.de).

number of samples, while for similar tasks, the DTL only uses a small number of samples to fine-tune the pre-trained model, then we can obtain a model with excellent performance in new tasks. The DTL has achieved great success in many fields such as computer vision (CV) and natural language processing (NLP) domains [32], [34], [35]. In the field of wireless communications, the DTL is applied in spectrum prediction and fault diagnosis [36], [37]. In [38], the DTL is applied to FDD massive MIMO systems for downlink CSI prediction. In fact, downlink CSI prediction and downlink CSI feedback are two different schemes. Both schemes ensure that the BS has access to full downlink CSI. In downlink CSI prediction scheme, the BS exploits the uplink CSI to directly predict the downlink CSI without feedback overhead. However, in downlink CSI feedback scheme, the BS recovers the downlink CSI from the compressed codeword sent by the user equipment (UE) with low feedback overhead.

In this paper, we employ the DTL based on a fully convolutional network architecture to cope with the high training cost of CSI feedback network. Specifically, we propose a DTL-based algorithm for the downlink CSI feedback in FDD massive MIMO systems, and discuss a tradeoff between training cost and model performance. Moreover, to reduce the requirement of a large training samples to pre-train a DNN, we propose a model-agnostic meta-learning (MAML) [39] based algorithm, which only requires relatively small number of samples to learn a model initialization. In the experimental evaluation, the performance of the model that trained with large number of samples and that of the model that fine-tuned with a relatively small number of samples is compared to demonstrate the effectiveness and superiority of DTL. At the same time, on the premise that a certain performance loss compared with the DTL algorithm, the MAML algorithm also shows good performance in terms of the NMSE. Our major contributions are summarized as follows.

- **DTL-based downlink CSI feedback:** We apply the DTL to the downlink CSI feedback, and propose a DTL-based algorithm to solve the problem of high training cost of CSI feedback network. By fine-tuning the pre-trained model using small number of samples of a wireless channel environment, we can quickly obtain a model with good performance at low data cost and time cost. To the best of our knowledge, no paper has so far reported DTL for downlink CSI feedback in FDD massive MIMO systems.
- **A MAML-based algorithm:** In order to solve the problem of large number of samples required for the pre-trained model in a wireless channel environment, a MAML-based algorithm, which employs samples from multiple wireless channel environments to learn a model initialization was developed. The numerical results show good performance in terms of NMSE of the MAML-based algorithm.

II. System Model

In FDD systems, the process of downlink CSI feedback by using DNN is shown in Fig. 1. At the UE side, when the UE receives the pilot sent by the BS, it uses the pilot to work out the downlink CSI. In this case, the downlink CSI is a high-dimensional matrix, and it is then inputted into the encoder of DNN

$$s = f_{en}(H) \tag{1}$$

which encodes the downlink CSI matrix H into a low dimensional codeword s. Next, the codeword is sent to the BS to complete the downlink CSI feedback. After the BS receives the codeword from the UE, the codeword is then inputted into the decoder of DNN

$$H = f_{de}(s) \tag{2}$$

which decodes the codeword s back to the original downlink CSI matrix H, and then the BS uses the downlink CSI to implement various measures to maintain high quality of wireless communication.

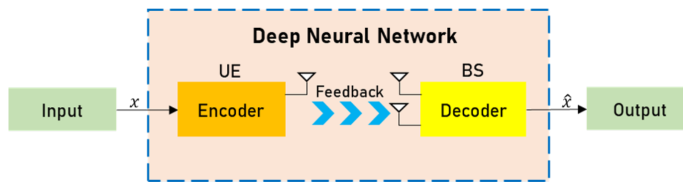

Fig. 1. A process of downlink CSI feedback.

The fifth generation (5G) new radio (NR) supports the spectrum ranging from sub-1 GHz to millimeter wave bands. In 3GPP R15, there are two frequency ranges (FR): the $0.45 - 6$ GHz is defined as FR1, commonly referred to as sub-6 GHz, while the $24.25 - 52.6$ GHz is defined as FR2, commonly referred to as millimeter wave [40]. In the frame structure of 5G NR, a frame has a duration of 10 ms, which consists of 10 subframes, and each subframe consists of $2^k \cdot 14$ $(k = 0, 1, 2, 3, 4)$ orthogonal frequency division multiplexing (OFDM) symbols. Correspondingly, the subcarrier spacing is $2^k \cdot 15$ kHz.

We consider adopting an OFDM model in FDD massive MIMO systems. The BS is equipped with N_b antennas, the UE is equipped with N_u antennas, and the number of subcarriers is N_s. Hence, a CSI sample containing one time slot is a complex matrix with size of $N_b \times N_u \times N_o \times N_s$, where the N_o represents the number of OFDM symbols of a time slot.

III. Problem Formulation

In this section, we first give the definition of DTL. Next, we address the DTL problem of downlink CSI feedback in FDD massive MIMO systems and propose a corresponding DTL model.

A. Definition of DTL

There are two basic concepts in transfer learning, i.e., domain and task. The domain is defined as $D = \{\chi, P(X)\}$, which is composed of the feature space χ and the marginal probability distribution $P(X)$, where $X = \{x_1, \cdots, x_n\} \in \chi$. The task is defined as $T = \{y, f(\cdot)\}$, which is composed of the label space y and the target prediction function $f(\cdot)$, where $f(\cdot)$ can be learned from training data. For a specific sample x, $f(x)$ can be written as the conditional probability distribution $P(y|x)$ from the probabilistic view. According to [32], transfer learning can be defined as:

***Definition 1 (Transfer learning)*:** Given a source domain D_S and source task T_S, a target domain D_T and target task T_T, transfer learning aims to improve the learning of the target prediction function $f_T(\cdot)$ of T_T by using the knowledge in D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$.

DNN has a strong transferability. Transfer learning can learn feature expressions unrelated to the domain, thus it can be combined with deep learning, and use the DNN to learn the common feature expressions of all domains. Based on [33], DTL can be defined as follows:

***Definition 2 (Deep transfer learning)*:** Given a transfer learning task defined by $\langle D_S, T_S, D_T, T_T \rangle$, it is a DTL task when the target prediction function $f_T(\cdot)$ of T_T is a non-linear function approximated by a deep neural network.

B. DTL Problem of CSI Feedback

FDD massive MIMO systems consider the spatial characteristics of the wireless channel, hence it is necessary to apply a new channel model to simulate the wireless environments in existing network. 3GPP R15 TR38.901 defines a new channel model named clustered delay line (CDL) for link evaluation, where the frequency range of the CDL model is 100 GHz with a maximum bandwidth of 2 GHz [41]. The CDL model is often implemented by phase initialization along four different polarizations and generating coefficients for each cluster [41]. The CDL model is divided into the CDL-A, CLD-B, CDL-C, CDL-D and CDL-E according to the simulated network environments. The first three are used to simulate non-line-of-sight (NLOS) transmission channels, while the latter two are used to simulate line-of-sight (LOS) transmission channels. To achieve excellent CSI feedback in a certain channel, we consider employing large number of samples of the channel to train a DNN from scratch. However, training the DNN this way for each channel which results in high training cost, both in data and time. Inspired by the image processing domain, the lower convolutional layers of DNN capture low-level image features, which are invariant in different tasks, while the higher convolution layers capture more complex details. Therefore, we propose to adopt DTL to extract the channel-independent correlations in the downlink CSI matrix to help train the DNNs of other channels, thereby solving the problem of high training cost.

Fig. 2 shows the DTL model for downlink CSI feedback proposed in this paper. First of all, large number of samples of CDL-A channel are used to train a DNN as the pre-trained model, and then the pre-trained model is fine-tuned with small number of samples of CDL-B, CDL-C, CDL-D, and CDL-E channels, respectively. Thus, the downlink CSI feedback model of different channels can be obtained at low training cost.

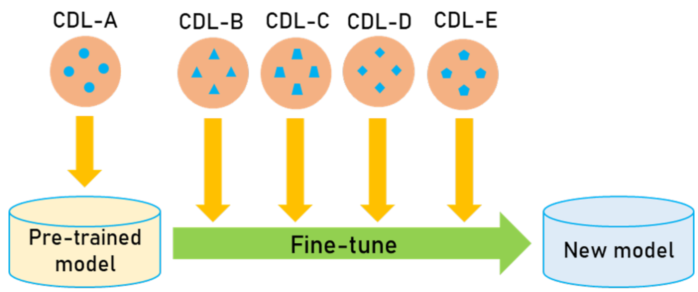

Fig. 2. The DTL model for downlink CSI feedback.

IV. The Proposed DTL Algorithm for CSI Feedback

Based on the DTL model for downlink CSI feedback introduced in Section II, we propose a DTL-based algorithm for downlink CSI feedback in FDD massive MIMO systems. In this section, we will specifically introduce the network structure, algorithm and optimization process used in DTL.

A. Network Architecture

To compress and recover the downlink CSI, the fully connected layers are used in the output of the encoder and the input of the decoder in CsiNet [26]. Different from the CsiNet which uses the fully connected layers, a fully convolutional network is used to reduce the number of network parameters in this paper, which adopts the network structure of the CsiNet. At the same time, the depth of the encoder network is increased. Moreover, the 3D convolution is used to replace the 2D convolution for adapting the dataset used in experiments. The fully convolutional network structure is shown in Fig. 3, where the black numbers represent the number of convolutional kernels. As can be seen from Fig. 3, the fully convolutional network is composed of input layer, encoder, decoder and output layer. The encoder includes eight convolutional layers, and the decoder includes one deconvolutional layer and three residual blocks, among which the three residual blocks have the same structure. They all consist of three convolutional layers, with a total of nine convolutional layers. It should be pointed out that Fig. 3 shows the network structure when the compression ratio is $1/8$. If we need to alter the compression ratio of the network, we can increase the convolutional layer and the deconvolutional layer in the output of the encoder and the input of the decoder, respectively. We achieve data compression and recovery by setting the convolutional kernel strides of the last convolutional layer of the encoder and the deconvolutional layer of the decoder, while the convolutional kernel strides of the rest convolutional layers are set as 1 to remain the dimension of data unchanged. An original sample of downlink CSI is a complex matrix with size of $N_b \times N_u \times N_o \times N_s$. To perform 3D convolution, the sample is preprocessed by the reshape function transforming into a complex matrix with size of $N_b \times (N_u \times N_o) \times N_s$. For the convolutional layers of the DNN, we use the leaky rectified linear unit (LeakyRelu) as an activation function, whose mathematical expression is defined as:

$$f(x) = \begin{cases} x, & x \geq 0 \\ x/a, & x < 0 \end{cases} \tag{3}$$

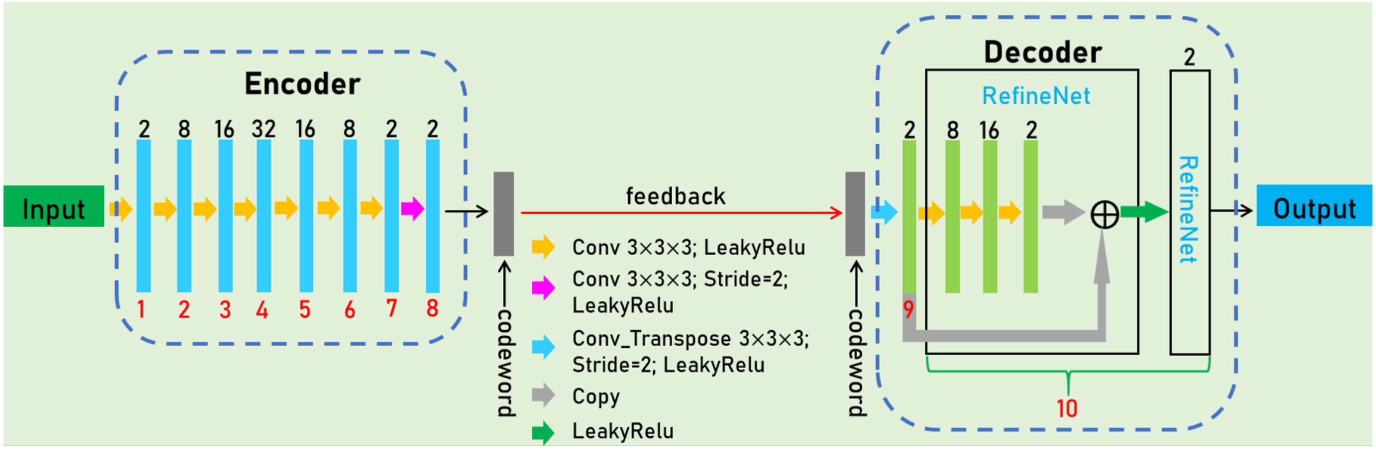

Fig. 3. Structure of the fully convolutional neural network.

where a is a fixed parameter in $[1, +\infty]$. Mean square error (MSE) is used as the loss function of the DNN, which is defined as follows,

$$L(x, \hat{x}) = \frac{1}{m} \sum_{i=1}^{m} \|x - \hat{x}\|_2^2 \tag{4}$$

where x and $\hat{x}$ represent the input and output of the DNN, respectively, m represents the number of samples in the training set, and $\|\cdot\|_2$ denotes the L_2 norm.

B. The DTL Algorithm

Machine learning and deep learning are modeling, training and testing in independent domains, while transfer learning tries to transfer knowledge from the source domains to the target domains, which makes the target domains achieve better learning effects. In general, the dataset of the source domains is abundant, while the dataset of the target domains is small. In this paper, for the target domains, namely the new wireless channel environments, we cannot always obtain large number of samples, and even if we can, it will take huge data cost and time cost to train a DNN from scratch. Since a model with good performance can be obtained by fine-tuning the pre-trained model using small number of samples on similar tasks in transfer learning, we combine the deep learning and transfer learning, and propose a DTL algorithm to solve above problems. The specific algorithm flow is shown in **Algorithm 1**.

Algorithm 1: The proposed DTL algorithm for downlink CSI feedback in FDD massive MIMO systems.

Input: Network parameters of the pre-trained model: θ_A, network architecture, CDL-B, CDL-C, CDL-D and CDL-E dataset, number of the gradient descents for fine-tuning: epoch, optimizer: Adam, learning rate: α, batch size: V
Output: Network parameters: $\{\theta_B, \theta_C, \theta_D, \theta_E\}$, NMSE and test loss
Training stage;
for k = *CDL-B, CDL-C, CDL-D, CDL-E* **do**
    Load the network parameters θ_A into network architecture;
    for $i = 1, 2, \cdots, epoch$ **do**
        Update the network parameters using the training set $D_{tr}(k)$ and the Adam optimizer with learning rate α
    end
    Save the network parameters θ_k;
end
Testing stage;
for k = *CDL-B, CDL-C, CDL-D, CDL-E* **do**
    Load the network parameters θ_k into network architecture;
    Predict the downlink CSI, calculate the NMSE and test loss using the testing set $D_{te}(k)$
end

C. Transfer Strategy and Improvement

In DTL, in order to obtain the optimal model performance of a target domain, all the samples of the target domain are generally used to fine-tune the pre-trained model. However, the requirement for real-time performance is very high in communication systems. In FDD massive MIMO systems, the wireless channel environment changes very quickly. It is possible that by the time we obtain a model with optimum performance, the UE is already in a new wireless channel environment, and the model is not effective at this time. Therefore, our goal is to obtain a model with good performance as soon as possible, instead of pursuing the optimum performance of model, as long as the model performance can guarantee the quality of wireless communication. Hence, we plan to explore the effects of sample sizes used for fine-tuning on model performance, and try to reduce the number of samples used in DTL to reduce the training cost in training process. In addition, the performance of the model obtained by fine-tuning the whole network is generally better than that of the model obtained by fine-tuning the part of network, but we consider whether it is possible that only some layers of network are fine-tuned, while the remaining layers of network are frozen.

Thus, we also intend to study the effects of different layers on the model performance, and further reduce the training cost by controlling the number of network parameters that needed to be fine-tuned on the premise that the model performance meets the actual application requirements.

V. The Proposed MAML Algorithm for CSI Feedback

In this section, we first give the definition of MAML and introduce the computational process. Next, we display the specific procedure of the proposed MAML algorithm.

A. Definition of MAML

Meta-learning is an exciting research direction in the field of machine learning, as it solves the problem of learning how to learn. The network structure, model initialization method, parameter updating model and so on can be designed by meta-learning algorithms. MAML is the simplest form of the meta-learning. In MAML, the network structure and parameter updating model are fixed, and it only learns a model initialization that can quickly adapt to a new task with small number of samples. Different from the DTL that requires large number of samples to train a DNN for obtaining the pre-trained model, the MAML needs multiple tasks to learn a model initialization as the pre-trained model, where each task only provides small number of samples. In addition, the MAML is independent of specific model, and the only requirement is to use the gradient descent algorithm to update the parameters. Therefore, the MAML can be applied to multiple learning problems, such as regression, classification and reinforcement learning. The MAML is composed of meta-training stage and meta-testing stage. In the meta-training stage, two datasets called support set and query set are used for learning a model initialization, while the training, validation and testing sets are used in the meta-test stage for fine-tuning and evaluation.

We consider the parameters of a DNN are initialized as ϕ, which become $\hat{\theta}^n$ after training on the support set of the nth task, and then the loss on the query set of the nth task is $l_n(\hat{\theta}^n)$. Thus, the loss function of the initialization parameters ϕ can be written as:

$$L(\phi) = \sum_{n=1}^{N} l_n(\hat{\theta}^n) \tag{5}$$

where N represents the number of tasks. Next, we think of getting parameters ϕ^*, which satisfies the following formula:

$$\phi^* = \arg\min_{\phi} L(\phi) \tag{6}$$

The gradient descent algorithm is adopted, and then the parameters ϕ can be updated as:

$$\phi \leftarrow \phi - \alpha \nabla_{\phi} L(\phi) \tag{7}$$

where α represents the learning rate. To compute the result more quickly, there are two main computational adjustments in MAML. First, it will take longer to conduct multiple steps of gradient descent for updating parameters from ϕ to $\hat{\theta}^n$ in each training task, thus only one step of gradient descent is conducted in MAML. The reason the MAML adopts one step of gradient descent is that if the model trained by one step of gradient descent achieves good performance, then the initialization parameters ϕ can basically be considered as good parameters. The second adjustment of MAML is to simplify the computation of $\nabla_{\phi} L(\phi)$. Because of a second order differential in $\nabla_{\phi} L(\phi)$, and it is difficult to compute, hence the second order differential is discarded in MAML, which is called first-order approximation [39].

B. The MAML Algorithm

Large number of samples of CDL-A channel are used to pre-train a model in DTL. However, in practice, it is difficult for us to obtain enough samples of a wireless channel environment to pre-train a model. On the contrary, it is relatively easy to obtain a mixed large dataset, which is composed of samples from multiple wireless channel environments, where each wireless channel environment only provides small number of samples. Thus, we propose a MAML algorithm for downlink CSI feedback in FDD massive MIMO systems. The specific algorithm procedure is shown in **Algorithm 2**. Moreover, in order to prevent gradient explosion, the gradient clipping is used in the meta-training stage, which is defined as:

$$grad \leftarrow grads \cdot \frac{clip_{norm}}{\max(global_{norm}, clip_{norm})} \tag{8}$$

where the $grads$ and $global_{norm}$ represent the gradients and the L_2 norm of the gradients, respectively, while the $clip_{norm}$ represent the ratio of clipping.

VI. Experimental Results

In this section, we first introduce the generation of dataset and the setting of experimental parameters. Then we show the comparison of model performance. Finally, we discuss the effects of different layers on model performance and training cost.

A. Dataset Generation and Parameters Setting

In the experiment, we consider a FDD massive MIMO scenario. To generate the corresponding downlink CSI dataset, we refer to the NR CDL channel model defined in TR38.901 of 3GPP R15, and use the NR CDL channel model of 5G toolbox in MATLAB software to generate the required downlink CSI dataset, namely, CDL-A, CDL-B, CDL-C, CDL-D and CDL-E dataset. For each channel, four speeds are used in the UE, which are 4.8 km/h, 24 km/h, 40 km/h and 60 km/h, respectively. The uplink and downlink frequencies of the channel are set to 2 GHz and 2.1 GHz, respectively. The number of subcarriers is set as 72 with a spacing of 15 KHz. The uniform plane array (UPA) is used at the BS with 32 antennas, while the UE is equipped with 2 antennas. The number of OFDM symbols is set as 14. Therefore, a downlink CSI sample is a complex matrix with size of $32 \times 2 \times 14 \times 72$. The training, validation, and testing sets of the CDL-A channel contain $50,000$, $5,000$, and $5,000$ samples, respectively, while those of the other channels used for DTL contain $4,000$, $1,000$, $5,000$ samples, respectively. The reason we set the sample

Algorithm 2: The proposed MAML algorithm for downlink CSI feedback in FDD massive MIMO systems.

Input: Network architecture, task set: {CDL-A, CDL-B, CDL-C, CDL-D and CDL-E}, iteration times of the meta-training: iterations, number of the gradient descents for fine-tuning: epoch, inner learning rate: α, meta learning rate: β, adam learning rate: η, batch size: V

Output: Pre-trained model, network parameters: $\{\theta_A, \theta_B, \theta_C, \theta_D, \theta_E\}$, NMSE and test loss

Meta-training stage;

Randomly initialize the network parameters as ϕ;

Generate corresponding support set and query set;

for $i = 1, 2, \cdots$, iterations **do**

  Randomly sample a task T from the task set;

  Randomly select V samples from the support set of the task T;

  Compute the gradient $\triangledown_{\phi} l(\phi)$ with the V samples;

  Update the network parameters from ϕ to $\hat{\theta}$: $\hat{\theta} = \phi - \alpha \triangledown_{\phi} l(\phi)$;

  Randomly select V samples from the query set of the task T;

  Compute the loss function $l(\hat{\theta})$ and the gradient $\triangledown_{\phi} l(\hat{\theta})$ with the V samples;

  Update the network parameters ϕ: $\phi = \phi - \beta \triangledown_{\phi} l(\hat{\theta})$;

end

Save the network parameters ϕ;

Meta-testing stage;

for k = *CDL-A, CDL-B, CDL-C, CDL-D, CDL-E* **do**

  Load the network parameters ϕ into the network architecture;

  for $i = 1, 2, \cdots$, epoch **do**

    Update the network parameters using the training set $D_{tr}(k)$ and the Adam optimizer with learning rate η

  end

  Save the network parameters θ_k;

  Predict the downlink CSI, calculate the NMSE and test loss using the testing set $D_{te}(k)$

end

size of the test sets of different wireless channel environments to the same is to demonstrate the effectiveness and superiority of DTL. That is, on the premise that there is an order of magnitude difference in the sample size of different training sets, if the model obtained by DTL and the CDL-A model have similar performance when evaluated on the testing set of the same sample size, then the DTL can be proved effective. Throughout all experiments, a relatively reasonable setting of hyperparameters is adopted. The epoch and batch size are set as 200 and 50, respectively. The learning rate is set as 0.001 in the training stage of the CDL-A model, and 0.0001 in the DTL stage. While in the MAML algorithm, for each channel, the total sample size of support set and query set is $4,000$ in meta-training stage. In meta-testing stage, the sample sizes of the training, validation and testing sets are set to $3,500$, $1,500$ and $1,000$, respectively. The relevant parameters of the MAML algorithm are listed in Table I. Default setting is adopted for other unspecified parameters. All the experiments are implemented on NVIDIA GeForce GTX 1080 Ti GPU.

TABLE I
PARAMETERS SETTING OF THE MAML ALGORITHM.

Parameters	Values
Ratio of support set	0.7
Ratio of query set	0.3
Inner learning rate α	0.001
Meta learning rate β	0.001
Adam learning rate η	0.0001
Batch size of meta-training stage	20
Batch size of meta-testing stage	50
Number of iterations	20000

The normalized MSE (NMSE) is used to evaluate the model performance, which is defined as follow:

$$NMSE = E\left[\left\|H - \hat{H}\right\|_2^2 / \|H\|_2^2\right] \tag{9}$$

where H represents the original CSI matrix, and $\hat{H}$ represents the recovered CSI matrix. To perform following experiments, we give the definition of the compression ratio γ, which can be written as follows:

$$\gamma = M/N \tag{10}$$

where M represents the output size of the encoder, and N represents the size of an original downlink CSI sample, i.e., $N = N_b \times N_u \times N_o \times N_s$.

B. Performance Comparison Under Different Wireless Channel Environments

We conduct experiments under four different compression ratios. For each compression ratio γ, the NMSE performance of five models corresponding to five different channels are compared. The experimental results are shown in Table II, where the performance of the CDL-A model is obtained by training the DNN from scratch with $50,000$ samples, while that of the other channel models is obtained using $4,000$ samples to fine-tune the CDL-A pre-trained model.

As can be seen from Table II, for $\gamma = 1/8$, the NMSE of the CDL-A model is -28.449 dB, while the NMSE of the CDL-B and CDL-C models are similar to that of the CDL-A model, which are -26.934 dB, -29.066 dB, respectively. The NMSE of the CDL-D and CDL-E models is even better than that of the CDL-A model, which are -33.646 dB and -33.532 dB, respectively. As for the reason, we consider it is due to the CDL-A, CDL-B and CDL-C are NLOS channels, while the CDL-D and CDL-E are both LOS channels, which are not as complex as the NLOS channels. When the γ is gradually reduced from $1/8$ to $1/256$, the NMSE of the CDL-A model is also gradually decreased from -28.449 dB to -12.887 dB. However, in different compression ratios, the performance of the CDL-B and CDL-C models is only slightly decreased compared with that of the CDL-A model, while the performance of the CDL-D and CDL-E models is better than that of the CDL-A model. In four different compression ratios,

TABLE II
PERFORMANCE COMPARISONS OF DIFFERENT MODELS.

γ	Channel model	NMSE (dB)	Test loss
1/8	CDL-A	-28.449	5.72×10^{-4}
	CDL-B	-26.934	8.53×10^{-4}
	CDL-C	-29.066	6.14×10^{-4}
	CDL-D	-33.646	3.07×10^{-4}
	CDL-E	-33.532	3.12×10^{-4}
1/64	CDL-A	-16.940	7.62×10^{-3}
	CDL-B	-13.565	1.79×10^{-2}
	CDL-C	-15.553	1.31×10^{-2}
	CDL-D	-23.487	3.14×10^{-3}
	CDL-E	-23.177	3.36×10^{-3}
1/128	CDL-A	-16.109	8.82×10^{-3}
	CDL-B	-12.887	2.10×10^{-2}
	CDL-C	-14.784	1.56×10^{-2}
	CDL-D	-21.993	4.44×10^{-3}
	CDL-E	-22.077	4.34×10^{-3}
1/256	CDL-A	-12.431	2.05×10^{-2}
	CDL-B	-8.454	5.84×10^{-2}
	CDL-C	-9.860	4.81×10^{-2}
	CDL-D	-19.381	8.05×10^{-3}
	CDL-E	-18.299	1.03×10^{-2}

the training time of the CDL-A model is about 50h, while that of the other channel models is about 4h20min. Hence, compared with the CDL-A model, the models of other channel environments can be obtained with only about $1/10$ of the data and time cost of that of the CDL-A model. At the same time, their performances are comparable to that of the CDL-A model, which implies that the DTL can well solve the problem of high training cost of CSI feedback network.

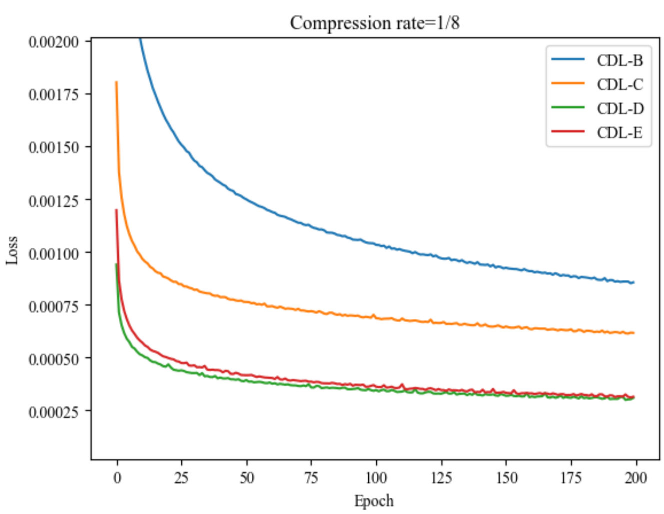

Fig. 4. Loss of the training set during fine-tuning process.

Fig. 4 shows the loss curve of the training set that used to fine-tune the pre-trained model. As shown in this figure, the losses of the CDL-D and CDL-E models decrease rapidly, and then begin to flatten when the epoch is about 25. While the losses of the CDL-B and CDL-C models decrease slowly compared with those of the CDL-B and CDL-C models, they still converge quickly. When the epoch reaches 200, although the losses of the CDL-B and CDL-C still have room to decline, there are limited improvements for model performance. From the four curves, it can be seen that the loss converges quickly when performing DTL, which implies the DTL is effective, i.e., the model with excellent performance can be obtained in a very short time under a new channel environment. At the same time, it can be seen that the losses of the LOS channels decline faster and smaller than that of the NLOS channels. Therefore, we can reduce the training time of the LOS channels and increase the training time of the NLOS channels when performing DTL. The CDL-B and CDL-C are NLOS channels, and their models show similar performance under different compression rates. The CDL-D and CDL-E are LOS channels, and their models also show similar performance under different compression rates. Therefore, the following experiments only focus on the CDL-B and CDL-D channels.

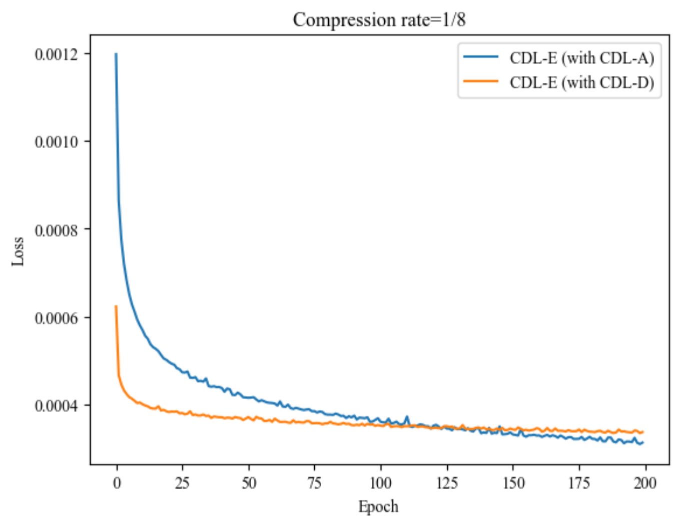

Fig. 5. Loss of the CDL-E model during fine-tuning process.

In DTL, it may be wrongly assumed that a better performance gain when transferring models between channels of the same type. In other words, we transfer the models of the NLOS channels to other NLOS channels, and transfer the models of the LOS channels to other LOS channels. However, this is not the case. The CDL-A channel is a NLOS channel, and the CDL-D and CDL-E channels are LOS channels. The CDL-A and CDL-D pre-trained models are trained from scratch with $50,000$ samples. The performance of the CDL-E model that fine-tuned with CDL-A pre-trained model is -33.532 dB, while that of the CDL-E model that fine-tuned with CDL-D pre-trained model is -33.084 dB, which is slightly decreased. Fig. 5 shows the loss of the CDL-E model during fine-tuning process. The CDL-E(with CDL-A) means that the CDL-E model is obtained by fine-tuning the CDL-A pre-trained model, while the CDL-E(with CDL-D) means that the CDL-E model is obtained by fine-tuning the CDL-D pre-trained model. From Fig. 5, we can see that the initial loss of the yellow curve is smaller than that of the blue curve and converges faster. However, after 200 epochs, the loss of the blue curve is smaller than that of the yellow curve, and there is still room to decline. We consider the reason is that the CDL-A is a NLOS channel while the CDL-D is a NLOS channel, and there is an inclusion relationship between them, that is, the CSI data of NLOS channels is more complex and contains more features than that of LOS channels. Therefore, compared with the models of LOS channels, we should consider the models of NLOS channels as the pre-trained models, in order to obtain better model performance.

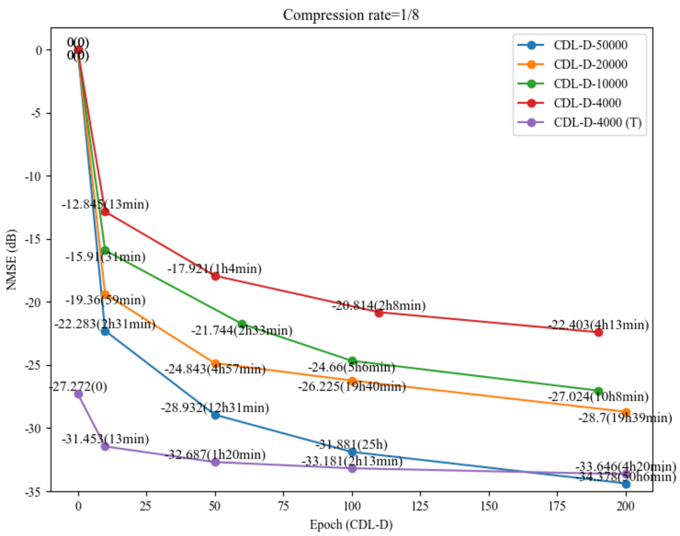

Fig. 6. Comparison of model performance.

Fig. 6 depicts the NMSE of models trained with different sample sizes during training process, where the epoch is 0, 10, 50, 100, 200, respectively. As shown in Fig. 6, the CDL-D-50000, CDL-D-20000, CDL-D-10000, and CDL-D-4000 represent that the DNN is trained from scratch with $50,000$, $20,000$, $10,000$ and $4,000$ samples, respectively, which does not involve DTL, and it is used for comparison here. The CDL-D-4000(T) represents that the DNN is fine-tuned with $4,000$ samples based on the CDL-A pre-trained model. The texts in the figure represent the NMSE and training time of model. As can be seen from Fig. 6, the NMSE of the four models trained from scratch is 0 dB when the epoch is 0, while the NMSE of the model fine-tuned with $4,000$ samples is -27.272 dB, which implies that it is effective to use the parameters of the pre-trained model to initialize the DNN. In addition, when we train the model from scratch, as the number of samples increases, the model performance on the testing set is also improving. At the end of the training, the NMSE of the CDL-D-50000 and the CDL-D-4000(T) is -34.378 dB and -33.646 dB, respectively, which are very similar, but with a huge difference of training time. It takes a training time of 50h6min to train a DNN from scratch, while it only takes 4h20min to perform fine-tuning, which indicates the effectiveness and superiority of the DTL. For the curves of the CDL-D-4000 and the CDL-D-10000, the reason we only show the NMSE of the model when epoch is 190 is that we set a modelcheckpoint of tensorflow during training process, and when the epoch increases from 190 to 200, the loss of model on the validation set is not decreased, hence the model is not saved when epoch is 200.

C. Performance Comparison in Different Sample Sizes

In previous experiments, $4,000$ samples are used to fine-tune the pre-trained model. Next, we explore the effects on model performance when different sample sizes are used to fine-tune the pre-trained model. In addition, γ is fixed as 1/8 for the following experiments. Table III shows the NMSE and training time, where the models are obtained by fine-tuning the pre-trained model using different sample sizes. From Table III, we can see that when the sample size reduces from $4,000$ to 200, the model performance gradually declines. The NMSE of the CDL-B model declines from -26.934 dB to -23.267 dB, and the NMSE of the CDL-D model declines from -33.646 dB to -31.392 dB. Nevertheless, with the reduction of the sample size, the training cost is also gradually reduced. When the model is fine-tuned with 200 samples, it only takes $1/20$ data cost and about $1/8$ time cost of those of the model fine-tuned with $4,000$ samples, with only a drop of performance about 3.5 dB in CDL-B channel, and about 2.5 dB in CDL-D channel. These results indicate that only a small number of samples are required to fine-tune the pre-trained model, and we can obtain a model that has good adaptability to the new wireless channel environment. From the above analysis, we can find that the reduction of sample size can further reduce the training cost by bearing a small loss of model performance.

TABLE III
COMPARISONS OF NMSE AND TRAINING TIME.

Sample size	CDL-B		CDL-D	
	NMSE (dB)	Training time	NMSE (dB)	Training time
4,000	−26.934	4h20min	−33.646	4h20min
3,000	−26.570	3h30min	−33.512	3h50min
2,000	−26.070	2h18min	−33.123	2h28min
1,000	−25.260	1h23min	−32.538	1h23min
500	−24.423	1h	−32.100	1h
200	−23.267	35min	−31.392	35min

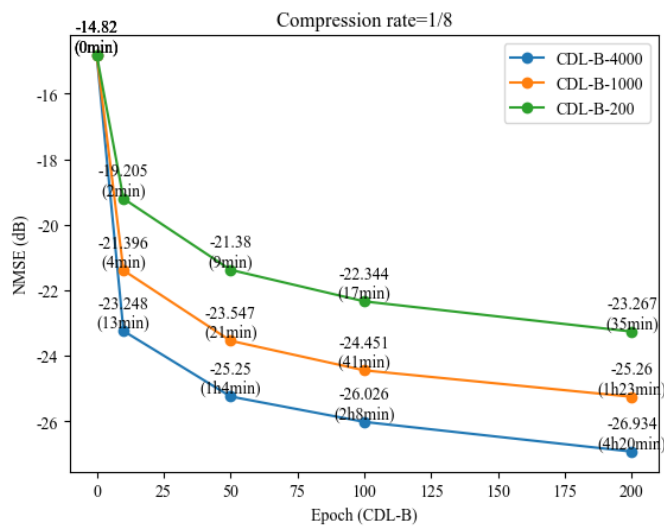

Fig. 7. Model performance of the fine-tuning process.

Fig. 7 displays the NMSE of the CDL-B model during fine-tuning process, where the model is fine-tuned with $4,000$, $1,000$ and 200 samples, respectively. As can be seen from Fig. 7 that the NMSE of the model fine-tuned with 200 samples and 200 epochs is similar to that of the model fine-tuned with $4,000$ samples and 10 epochs. Therefore, we can make a tradeoff between data cost and time cost. For example, if we set -23 dB as the baseline of the model performance, then the model needs to be fine-tuned with more epochs when there are few samples, while the model needs to be fine-tuned with less epochs when there are enough samples. In addition, due to the NMSE of the model fine-tuned with $1,000$ samples is better than that of the model fine-tuned with 200 samples, and whose training cost is less than that of the model fine-tuned with $4,000$ samples, $1,000$ samples are adopted to conduct the following experiments.

D. Effects of Different Layers on Model Performance

In previous experiments, the parameters of the pre-trained model are used to initialize the parameters of the new model, and then all the layers of the new model are fine-tuned using samples of a new wireless channel environment. In this section, in order to explore the effects of different layers on model performance, only some layers of the model are fine-tuned, while other layers are frozen. In Fig. 3, the red numbers represent the corresponding layers. The number 10 represents the three residual blocks of the DNN; the number 8 and number 9 represent the output layer of the encoder and the input layer of the decoder, respectively; the number 4 represents the convolutional layer with 32 convolutional kernels; the number $1 \sim 10$ represents all the layers of DNN.

TABLE IV
MODEL PERFORMANCE UNDER DIFFERENT TRAINING STRATEGIES.

Layer	NMSE (dB)		No. parameters	Training time
	CDL-B	CDL-D		
1-10	-25.260	-32.538	$50,170$	58mim
10	-22.340	-31.530	$14,334$	41min
8,9,10	-22.630	-31.558	$14,554$	42min
4	-16.686	-28.912	$13,856$	41min
4,10	-24.324	-32.397	$28,190$	53min

Table IV shows the comparison of the NMSE and training cost when $1,000$ samples are used to fine-tune the different layers of DNN. As can be seen from Table IV, when all the layers are fine-tuned, the NMSE of the CDL-B and CDL-D models is -25.260 dB and -32.539 dB, respectively, and the number of parameters that need to be trained is $50,170$ with a training time of 58min. When only the three residual blocks are fine-tuned, the NMSE of the CDL-B and CDL-D models is -22.340 dB and -31.530 dB, respectively, and there is a drop of performance about 3 dB and 1 dB, respectively. However, the number of parameters that need to be trained is only about $1/3$ of $50,170$, and the training time reduces from 58min to 41min. When the 8th, 9th and 10th layers are fine-tuned, it can be found that the model performance barely improves compared to that of only fine-tuning the 10th layer, which indicates that the gain of model performance is not very much when the 8th and 9th layers are added for fine-tuning. The number of parameters that need to be trained and the training time are similar to those of fine-tuning the 10th layer when only the 4th layer is fine-tuned, but the performance difference of the two models is obvious. There is about a performance loss of 6 dB in CDL-B channel, and 2.5 dB in CDL-D channel, which shows that fine-tuning the latter layers can obtain more performance gain than fine-tuning the former layers. In addition, compared with only fine-tuning the 10th layer, when the 4th and 10th layers are fine-tuned, the performance of the CDL-B and CDL-D models has a gain of 2 dB and 1 dB, respectively, but the number of parameters that need to be trained and training time will be greatly increased. Based on the above analysis, we consider only fine-tuning the 10th layer is a good scheme. In fact, the 10th layer is composed of three residual blocks with the same structure. Therefore, we next discuss the effects on model performance when different number of residual blocks are fine-tuned.

TABLE V
EFFECTS OF DIFFERENT RESNET BLOCKS ON MODEL PERFORMANCE.

Nmuber	NMSE (dB)		No. parameters	Training time
	CDL-B	CDL-D		
3	-22.340	-31.530	$14,334$	41mim
2	-20.587	-30.587	$9,556$	35min
1	-18.397	-29.276	$4,778$	29min

Table V shows the comparison of the NMSE and the training cost when different number of residual blocks are fine-tuned. It can be seen that when only the last residual block is fine-tuned, the NMSE of the CDL-B and CDL-D models is -18.379 dB and -29.276 dB, respectively. As the number of residual blocks that need to be fine-tuned increases, the model performance will increase linearly. In CDL-B and CDL-D channels, when one more residual block is added for fine-tuning, the model performance has a gain of about 2 dB and 1 dB, respectively, but the parameters that need to be fine-tuned and the training time will also increase linearly. From the above analysis, we can see that the three residual blocks have a great effect on the model performance. Therefore, we should first consider fine-tuning the three residual blocks when performing DTL.

E. Model Performance of the MAML Algorithm

Fig. 8 shows the NMSE of different models, where the pre-trained model is obtained by the MAML algorithm, and it is fine-tuned with $1,000$, $2,000$ and $3,000$ times, respectively. The ratio of support set is 0.7. As can be seen from Fig. 8, compared with the pre-trained model, the model performance gradually rises with the increase of the fine-tuning times. However, there is little performance gain when the fine-tuning times increases from $2,000$ to $3,000$, and when the fine-tuning times is 3,000, the model performance under different channels is not as good as well. Hence, the hyperparameters are adjusted to further improve model performance.

The experimental results of different ratios of support set are shown in summarized in Fig. 9. The pre-trained model 1 is obtained by setting the ratio of support set as 0.7 during the training process, while the pre-trained model 2 is obtained by setting the ratio of support set as 0.5. The $3,000$ represents the model obtained by fine-tuning the pre-trained model 1 with $3,000$ times. Similarly, the $1,000$ represents the model obtained by fine-tuning the pre-trained model 2 with $1,000$ times. From Fig. 9, we can see that when the ratio

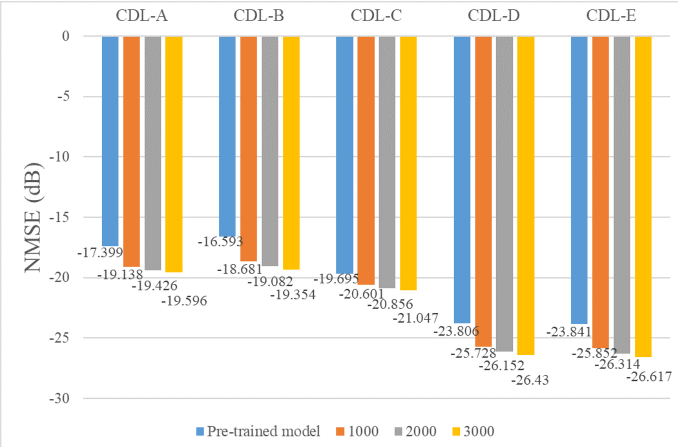

Fig. 8. NMSE of model under different fine-tuning times.

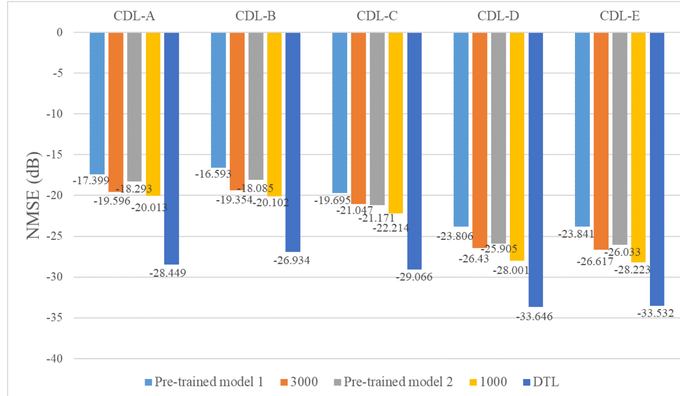

Fig. 9. NMSE of model under different ratios of support set.

of support set changes from 0.7 to 0.5, the performance of pre-trained model is improved under different channels, and the performance of the model fine-tuned with $1,000$ times is better than that of the model fine-tuned with $3,000$ times. However, there is still a certain performance loss compared with the DTL algorithm. The DTL in Fig. 9 represents the performance of the model that is pre-trained and fine-tuned with DTL algorithm. Compared with the the DTL algorithm, the best performance of the MAML algorithm has a drop about 8 dB in CDL-A channel, 7 dB in CDL-B and CDL-C channels and 5 dB in CDL-D and CDL-E channels. In addition, take the CDL-D wireless channel environment as an example, the performance of the CDL-D model trained from scratch with $4,000$ samples is -22.403 dB, which is shown in Fig. 6. However, the performance of the pre-trained 2 obtained by using the MAML algorithm among which each wireless channel environment provides $4,000$ samples is -25.905 dB, which is shown in Fig. 9. As for the reason that the performance of the latter model is better than that of the former model, we consider that in a small data set of a certain wireless channel environment, some common features that existed in each wireless channel environment can not be learned by the DNN when the model is trained from scratch. However, the data sets of different wireless channel environments are used in the MAML algorithm, and then the DNN can learn these common features. Hence, when there are only a few samples in each wireless channel environment, compared to using the samples in each wireless channel environment to train their respective models, we can obtain a pre-trained model that has better performance in each wireless channel environment by using the samples from multiple wireless channel environments and the MAML algorithm. Based on the above analysis, if we can get samples of a wireless channel environment as much as possible, it is better to adopt the DTL algorithm to obtain a pre-trained model and perform fine-tuning. On the contrary, we can adopt the MAML algorithm with a certain performance loss when we can only obtain a relatively small dataset, which is composed of samples from multiple wireless channel environments, where each wireless channel environment provides small number of samples.

Table VI displays the training cost of the DTL and MAML algorithms. In the process of training the pre-trained model, the batch size of the DTL algorithm is set as 50 with GPU memory usage 4433 M, and the batch size of the MAML algorithm is set as 20 with GPU memory usage 8531 M. After the DNN converges, the DTL algorithm takes a training time of 50h12min, while the MAML algorithm takes a training time of 15h42min. Therefore, compared with the DTL algorithm, the MAML algorithm has a higher computational complexity due to the high GPU memory usage, but it has lower time cost due to the low training time.

TABLE VI
COMPARISON OF TRAINING COST BETWEEN THE DTL AND MAML ALGORITHMS.

Algorithm	Batch size	GPU memory usage	Training time
DTL	50	4433M	50h12min
MAML	20	8531M	15h42min

VII. CONCLUSION

In this paper, we proposed a DTL-based method to solve the problem of high training cost associated with downlink CSI feedback network in FDD massive MIMO systems. Then, we discussed the effects of different layers, seeking a tradeoff between training cost and model performance. Furthermore, we proposed a MAML-based method to solve the problem related to large number of samples of a wireless channel environment that are required to train a DNN as pre-trained model. Experimental results proved the effectiveness and superiority of these proposed methods in reducing training cost. In addition, although the CDL channel model is used in the experiment of DTL, we can still apply DTL for other channel models to solve the problem of high training cost.

REFERENCES

[1] S. Qiu, Da Chen, D. Qu, K. Luo, and T. Jiang, “Downlink precoding with mixed statistical and imperfect instantaneous CSI for massive MIMO systems,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3028–3041, Apr. 2018.

[2] X. Liu, J. Zhang, and S. Cai, “An optimal power allocation scheme in downlink multi-user NOMA beamforming system with imperfect CSI,” in *IEEE International Conference on Communication Systems (ICCS)*, Chengdu, China, 2018, pp. 99–103.

[3] J. Kim, J. Choi, and J. M. Cioffi, "Cooperative distributed beamforming with outdated CSI and channel estimation errors," *IEEE Trans. Commun.*, vol. 62, no. 12, pp. 4269–4280, Dec. 2014.
[4] X. Kuai, X. Yuan, Y.-C. Liang, "Message-passing based OFDM receiver for time-varying sparse multipath channels," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 10097–10101, Oct. 2018.
[5] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6169–6183, Dec. 2015.
[6] P. N. Alevizos, X. Fu, N. D. Sidiropoulos, Y. Yang, and A. Bletsas, "Limited feedback channel estimation in massive MIMO with non-uniform directional dictionaries," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5127–5141, Oct. 2018.
[7] Y. Han, T. Hsu, C. Wen, K. Wong, and S. Jin, "Efficient downlink channel reconstruction for FDD multi-antenna systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3161–3176, June 2019.
[8] Y. Han, Q. Liu, C. Wen, M. Matthaiou, and X. Ma, "Tracking FDD massive MIMO downlink channels by exploiting delay and angular reciprocity," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 1062–1076, Sept. 2019.
[9] M. Jian, F. Gao, Z. Tian, S. Jin, and S. Ma, "Angle-domain aided UL/DL channel estimation for wideband mmWave massive MIMO systems with beam squint," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3515–3527, July 2019.
[10] Z. Qin, X. Zhou, L. Zhang, Y. Gao, Y.-C. Liang, and G. Y. Li, "20 years of evolution from cognitive to intelligent communications," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 6–20, Jan. 2020.
[11] S. Hu, Y. Pei, P. P. Liang, and Y.-C. Liang, "Deep neural network for robust modulation classification under uncertain noise conditions," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 564–577, Jan. 2020.
[12] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies towards 6G," *IEEE Wireless Commun. Mag.*, vol. 27, no. 3, pp. 96–103, Jun. 2020.
[13] P. Liang, J. Fan, W. Shen, Z. Qin, G. Y. Li, "Deep learning and compressive sensing-based CSI feedback in FDD massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9217–9222, Aug. 2020.
[14] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.
[15] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network towards 6G: Machine-learning approaches," *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.
[16] Y. Lin, M. Wang, X. Zhou, G. Ding, and S. Mao, "Dynamic spectrum interaction of UAV flight formation communication with priority: A deep reinforcement learning approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 892–903, Mar. 2020.
[17] Y. Lin, Y. Tu, Z. Dou, L. Chen, and S. Mao, "Contour stella image and deep learning for signal recognition in the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, early access, doi: 10.1109/TCCN.2020.3024610.
[18] N. Ye, X-M. Li, H. Yu, L. Zhao, W. Liu, X. Hou, "DeepNOMA: A unified framework for NOMA using deep multi-task learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2208–2225, Apr. 2020.
[19] M. Liu, J. Yang, T. Song, J. Hu, and G. Gui, "Deep learning-inspired message passing algorithm for efficient resource allocation in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 641–653, Jan. 2019.
[20] X. Sun, G. Gui, R. Liu, Y. Li and Y. An, "ResInNet: A novel deep neural network with feature reuse for internet of things," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 679–691, Feb. 2019.
[21] Z. Qin, H. Ye, G. Y. Li, B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Feb. 2019.
[22] F. Sohrabi, K. M. Attiah, and W. Yu, "Deep Learning for Distributed Channel Feedback and Multiuser Precoding in FDD Massive MIMO," arXiv preprint arXiv:2007.065122020
[23] K. Kong, W-J Song, and M. Min, "Deep-learning-based precoding in multiuser MIMO downlink channels with limited feedback," arXiv preprint arXiv:2008.04147
[24] K. M. Attiah, F. Sohrabi, and W. Yu, "Deep Learning Approach to Channel Sensing and Hybrid Precoding for TDD Massive MIMO Systems," arXiv preprint arXiv:2011.10709
[25] J. Jang, H. Lee, S. Hwang, H. Ren, and I. Lee, "Deep learning-based limited feedback designs for MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 558–561, Apr. 2020.
[26] C. Wen, W. Shih and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, Oct. 2018.
[27] T. Wang, C. Wen, S. Jin and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 416–419, April 2019.
[28] Z. Liu, L. Zhang and Z. Ding, "Exploiting bi-directional channel reciprocity in deep learning for low rate massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 889–892, June 2019.
[29] Y. Liao, H. Yao, Y. Hua and C. Li, "CSI feedback based on deep learning for massive MIMO systems," *IEEE Access*, vol. 7, pp. 86810–86820, 2019.
[30] Q. Sun, Y. Wu, J. Wang, C. Xu and K. Wong, "CNN-based CSI acquisition for FDD massive MIMO with noisy feedback," *Electronics Lett.*, vol. 55, no. 17, pp. 963–965, 2019.
[31] Y. Jang, G. Kong, M. Jung, S. Choi and I. Kim, "Deep autoencoder based CSI feedback with feedback errors and feedback delay in FDD massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 833–836, June 2019.
[32] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[33] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning," *Proc. Int. Conf. Artificial Neural Networks*. Springer, 2018, pp. 270–279.
[34] S. J. Pan, I. W. Tsang, J. T. Kwok and Q. Yang, "Domain Adaptation via Transfer Component Analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
[35] V. M. Patel, R. Gopalan, R. Li and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, May 2015.
[36] F. Lin, J. Chen, J. Sun, G. Ding and L. Yu, "Cross-band spectrum prediction based on deep transfer learning," *China Commun.*, vol. 17, no. 2, pp. 66–80, Feb. 2020.
[37] W. Wu, M. Peng, W. Chen and S. Yan, "Unsupervised deep transfer learning for fault diagnosis in fog radio access networks," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8956–8966, Sept. 2020.
[38] Y. Yang, F. Gao, Z. Zhong, B. Ai, and A. Alkhateeb, "Deep transfer learning based downlink channel prediction for FDD massive MIMO systems," *IEEE Trans. Commun.*, early access, doi: 10.1109/TCOM-M.2020.3019077.
[39] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *Proc.34th Int. Conf. Machine Learning*, 2017, vol. 70, pp. 1126–1135.
[40] X. Lin et al., "5G New Radio: Unveiling the Essentials of the Next Generation Wireless Access Technology," *IEEE Communications Standards Magazine*, vol. 3, no. 3, pp. 30-37, September 2019, doi: 10.1109/MCOMSTD.001.1800036.
[41] E. C. Chukwu, U. S. Abdullahi, G. Koyunlu, J. Sanusi, G. Sani and I. A. Gangfada, "Performance Evaluation of Multiplexed 5G-New Radio Network Services of Different Usage Scenarios," *2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India*, 2020, pp. 335-342, doi: 10.1109/ICCES48766.2020.9138021.

Yu Wang (S'18) received his B.S. degree in Communication Engineering from Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China in 2019. He is currently pursuing the master's degree in communication and information engineering with the Nanjing University of Posts and Telecommunications, Nanjing, China. His research interest includes machine learning for wireless communications.

Jinlong Sun (M'18) Jinlong Sun (Member, IEEE) received the M.S. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 2014 and 2018, respectively. He is currently working as an assistant professor with Nanjing University of Posts and Telecommunications, Nanjing, China. His current research interests include signal processing for wireless communications, machine learning, and integrated navigation systems

Guan Gui (M'11–SM'17) received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2012. From 2009 to 2014, he joined the Tohoku University as a research assistant as well as a postdoctoral research fellow, respectively. From 2014 to 2015, he was an Assistant Professor in the Akita Prefectural University. Since 2015, he has been a professor with Nanjing University of Posts and Telecommunications, Nanjing, China. His recent research interests include artificial intelligence, deep learning, non-orthogonal multiple access, wireless power transfer, and physical layer security.

Dr. Gui has published more than 200 IEEE Journal/Conference papers and won several best paper awards, e.g., ICC 2017, ICC 2014 and VTC 2014-Spring. He received the IEEE Communications Society Heinrich Hertz Award in 2021, the Elsevier Highly Cited Chinese Researchers in 2020, the Member and Global Activities Contributions Award in 2018, the Top Editor Award of IEEE Transactions on Vehicular Technology in 2019, the Outstanding Journal Service Award of KSII Transactions on Internet and Information System in 2020, the Exemplary Reviewer Award of IEEE Communications Letters in 2017. He was also selected as for the Jiangsu Specially-Appointed Professor in 2016, the Jiangsu High-level Innovation and Entrepreneurial Talent in 2016, the Jiangsu Six Top Talent in 2018, the Nanjing Youth Award in 2018. He is serving or served on the editorial boards of several journals, including IEEE Transactions on Vehicular Technology, IEICE Transactions on Communications, Physical Communication, Wireless Networks, IEEE Access, Journal of Circuits Systems and Computers, Security and Communication Networks, IEICE Communications Express, and KSII Transactions on Internet and Information Systems, Journal on Communications. In addition, he served as the IEEE VTS Ad Hoc Committee Member in AI Wireless, Executive Chair of IEEE VTC 2021-Fall, Vice Chair of IEEE WCNC 2021, TPC Chair of PHM 2021, General Co-Chair of Mobimedia 2020, TPC Chair of WiMob 2020, Track Chairs of IEEE VTC 2020-Spring, ISNCC 2020 and ICCC 2020, Award Chair of IEEE PIMRC 2019, and TPC member of many IEEE international conferences, including GLOBECOM, ICC, WCNC, PIRMC, VTC, and SPAWC. He is an IEEE Senior Member.

Bamidele Adebisi received his Bachelor's degree in electrical engineering from Ahmadu Bello University Zaria, Nigeria, in 1999, and his Masters degree in advanced mobile communication engineering and Ph.D. degree in communication systems from Lancaster University, United Kingdom, in 2003 and 2009, respectively. He was a senior research associate with the School of Computing and Communication, Lancaster University, from 2005 to 2012. He joined Manchester Metropolitan University in 2012, where he is currently a Full Professor (Chair) of intelligent infrastructure systems. He is the current Vice Chair of IEEE TC-PLC; was General Chair, IEEE ISPLC'18, UK; Co-Chair, 6th IEEE Int'l Conference on Smart Grid Communications, 2015, Miami, US, etc. He is a Panel Member of the UK Engineering and Physical Sciences Research Council (EPSRC) Peer Review College, and an EU H2020 Expert Reviewer/ rapporteur. He has been part of multi-partner, multi-country, multi-million pounds projects as PI and Co-I. One of his projects with an SME received the 2020 UK Best Knowledge Transfer Partnership Project of the Year Awards. He has published over 140 peer-review papers and given several talks/panel discussions in the research areas of Internet of Things, smart cities, smart grids, communication systems and cyber physical systems. Bamidele is a Fellow of IET, a Fellow of Higher Education Academy and a Chartered Engineer.

Tomoaki Ohtsuki (SM'01) received the B.E., M.E., and Ph. D. degrees in Electrical Engineering from Keio University, Yokohama, Japan in 1990, 1992, and 1994, respectively. From 1994 to 1995 he was a Post Doctoral Fellow and a Visiting Researcher in Electrical Engineering at Keio University. From 1993 to 1995 he was a Special Researcher of Fellowships of the Japan Society for the Promotion of Science for Japanese Junior Scientists. From 1995 to 2005 he was with Science University of Tokyo. In 2005 he joined Keio University. He is now a Professor at Keio University. From 1998 to 1999 he was with the department of electrical engineering and computer sciences, University of California, Berkeley. He is engaged in research on wireless communications, optical communications, signal processing, and information theory. Dr. Ohtsuki is a recipient of the 1997 Inoue Research Award for Young Scientist, the 1997 Hiroshi Ando Memorial Young Engineering Award, Ericsson Young Scientist Award 2000, 2002 Funai Information and Science Award for Young Scientist, IEEE the 1st Asia-Pacific Young Researcher Award 2001, the 5th International Communication Foundation (ICF) Research Award, 2011 IEEE SPCE Outstanding Service Award, the 27th TELECOM System Technology Award, ETRI Journal's 2012 Best Reviewer Award, and 9th International Conference on Communications and Networking in China 2014 (CHINACOM '14) Best Paper Award. He has published more than 205 journal papers and 415 international conference papers.

He served as a Chair of IEEE Communications Society, Signal Processing for Communications and Electronics Technical Committee. He served as a technical editor of the IEEE Wireless Communications Magazine and an editor of Elsevier Physical Communications. He is now serving as an Area Editor of the IEEE Transactions on Vehicular Technology and an editor of the IEEE Communications Surveys and Tutorials. He has served as general-co chair, symposium co-chair, and TPC co-chair of many conferences, including IEEE GLOBECOM 2008, SPC, IEEE ICC 2011, CTS, IEEE GLOBECOM 2012, SPC, IEEE ICC 2020, SPC, IEEE APWCS, IEEE SPAWC, and IEEE VTC. He gave tutorials and keynote speeches at many international conferences including IEEE VTC, IEEE PIMRC, IEEE WCNC, and so on. He was Vice President and President of the Communications Society of the IEICE. He is a senior member and a distinguished lecturer of the IEEE, a fellow of the IEICE, and a member of the Engineering Academy of Japan.

Haris Gacanin (SM'13–F'20) received his Dipl.-Ing. degree in Electrical engineering from the University of Sarajevo in 2000. In 2005 and 2008, respectively, he received MSc and Ph.D. from Tohoku University in Japan. He was with Tohoku University from 2008 until 2010 first as Japan Society for the Promotion of Science (JSPS) postdoctoral fellow and later, as an Assistant Professor. He joined Alcatel-Lucent Bell (now Nokia Bell) in 2010 as a Physical-layer Expert and later moved to Nokia Bell Labs as Department Head. Since April 2020, he joined RWTH Aachen University. He is a head of the Chair for Distributed Signal Processing and co-director of the Institute for Communication Technologies and Embedded Systems.

His professional interests are related to broad areas of digital signal processing and artificial intelligence with applications in wireless communications. He has 200+ scientific publications (journals, conferences and patents) and invited/tutorial talks. He is a fellow of IEEE. He was a Distinguished Lecturer of IEEE Vehicular Technology Society and an Associate Editor of IEEE COMMUNICATIONS MAGAZINE, while he served as the editor of IEICE Transactions on Communications and IET Communications. He acted as a general chair and technical program committee member of various IEEE conferences. He is a recipient of several Nokia innovation awards, IEICE Communications Society Best Paper Award in 2021, IEICE Communication System Study Group Best Paper Award (joint 2014, 2015, 2017), The 2013 Alcatel-Lucent Award of Excellence, the 2012 KDDI Foundation Research Award, the 2009 KDDI Foundation Research Grant Award, the 2008 JSPS Postdoctoral Fellowships for Foreign Researchers, the 2005 Active Research Award in Radio Communications, 2005 Vehicular Technology Conference (VTC 2005-Fall) Student Paper Award from IEEE VTS Japan Chapter and the 2004 Institute of IEICE Society Young Researcher Award.

Hikmet Sari (F'95-LF'20) received the engineering diploma and the Ph.D. degree from ENST, Paris, France, and the Habilitation degree from the University of Paris XI. From 1980 to 2002, he held various research and management positions at Philips Research Laboratories, SAT, Alcatel, Pacific Broadband Communications, and Juniper Networks. From 2003 to 2016, he was a Professor and the Head of the Telecommunications Department, Suplec, and a Chief Scientist at Sequans Communications. He is currently a Professor with the Nanjing University of Posts and Telecommunications (NJUPT). Dr. Saris distinctions include the Andre Blondel Medal in 1995, the Edwin H. Armstrong Achievement Award in 2003, the Harold Sobol Award in 2012, as well as election to the Academia Europaea (Academy of Europe) and the Science Academy of Turkey in 2012. He was the Chair of the Communication Theory Symposium of ICC 2002, a Technical Program Chair of ICC 2004, a Vice General Chair of ICC 2006, a General Chair of PIMRC 2010, a General Chair of WCNC 2012, an Executive Chair of WCNC 2014, a General Chair of ICUWB 2014, a General Co-Chair of IEEE BlackSeaCom 2015, a Technical Program Chair of EuCNC 2015, an Executive Co-Chair of ICC 2016, a General Co-Chair of ATC 2016, an Executive Chair of ICC 2017, a General Co-Chair of ATC 2018, and a General Co-Chair of PIMRC 2019. He also chaired the Globecom and ICC Technical Content (GITC) Committee from 2010 to 2011, and was the Communications Society (ComSoc) Vice President for Conferences from 2014 to 2015, the Director for Conference Operations of ComSoc from 2018 to 2019, and the Vice President for Conferences of the IEEE France Section from 2017 to 2019. He is currently serving as a General Chair of WCNC 2021 and an Executive Co-Chair of GLOBECOM 2023. He served as an Editor for the IEEE Transactions on Communications from 1987 to 1991, an Associate Editor for the IEEE Communications Letters from 1999 to 2002, and a Guest Editor for several special issues of the IEEE Journal on Selected Areas in Communications, European Transactions on Telecommunications (ETT), and other journals. He served as a Distinguished Lecturer of ComSoc from 2001 to 2006, a member of the IEEE Fellow Evaluation Committee from 2002 to 2007, and a member of the IEEE Awards Committee from 2005 to 2007.