Embedding Model Based Fast Meta Learning for Downlink Beamforming Adaptation

Juping Zhang, Yi Yuan, Gan Zheng, *Fellow, IEEE,* Ioannis Krikidis, *Fellow, IEEE,* and Kai-Kit Wong, *Fellow, IEEE*

Abstract

This paper studies the fast adaptive beamforming for the multiuser multiple-input single-output downlink. Existing deep learning-based approaches assume that training and testing channels follow the same distribution which causes task mismatch, when the testing environment changes. Although meta learning can deal with task mismatch, it rely on labelled data and incur high complexity in the pre-training and fine tuning stages. We propose a simple yet effective adaptive framework to solve the mismatch issue, which trains an embedding model as a transferable feature extractor, followed by fitting the support vector regression. Compared to the existing meta learning algorithm, our method does not necessarily need labelled data in the pre-training and does not need fine-tuning of the pre-trained model in the adaptation. The effectiveness of the proposed method is verified through two well-known applications, i.e., the signal to interference plus noise ratio balancing problem and the sum rate maximization problem. Furthermore, we extend our proposed method to online scenarios in non-stationary environments. Simulation results demonstrate the advantages of the proposed algorithm in terms of both performance and complexity. The proposed design framework can also be applied to general radio resource management problems.

Index Terms

Meta-learning, online learning, embedding model, beamforming.

J. Zhang and G. Zheng are with the Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, LE11 3TU, UK (E-mail: {j.zhang3, y.yuan, g.zheng}@lboro.ac.uk).

Y. Yuan was with the Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, LE11 3TU, UK and is now with the Department of Electrical and Electronic Engineering, University of Surrey, Guildford, GU2 7XH, UK (E-mail: yy0007@surrey.ac.uk).

I. Krikidis is with the Department of Electrical and Computer Engineering, University of Cyprus, 1678 Nicosia, Cyprus (E-mail: krikidis@ucy.ac.cy).

K.-K. Wong is with the Department of Electronic and Electrical Engineering, University College London, London, WC1E 6BT, UK (Email: kai-kit.wong@ucl.ac.uk).

Part of this work was accepted in the IEEE International Conference on Communications (ICC), 2021, Montreal, Canada.

I. Introduction

Beamforming is one of the most promising multi-antenna techniques that can realize the antenna diversity gain and mitigate multiuser interference simultaneously. Optimizing beamforming weights is critical to fully reap its benefits and has been extensively studied in the literature for various objectives such as power minimization [1], [2], signal-to-interference-plus-noise ratio (SINR) balancing [3], and sum rate maximization [4]. However, most beamforming solutions are highly complex to implement and cannot meet the critical latency requirement in the fifth generation (5G) and beyond systems because they are iterative in nature and provide slow convergence.

Recently the deep learning technique has been proposed to address the complexity of beamforming design using the 'learning to optimize' framework [5]. It is based on the intuitive idea that the mapping from channel state to beamforming can be learned by training a neural network model in an offline manner, and then the beamforming solution can be directly predicted using the trained model in real time. This method shifts the complexity of real-time beamforming optimization to offline training and its potential has been demonstrated in solving a series of beamforming design problems [6]–[11]. A major drawback of the existing deep learning-based beamforming solutions is that they are restricted to static wireless environments, in which the training and testing channels follow the same distribution; this assumption is practically unrealistic. In practical wireless networks, the channel distribution may change due to high mobility (e.g., in vehicular networks) or unexpected perturbations of complex environments. Consequently, a well trained model based on the training data could cause unacceptable performance degradation in the testing environment. To tackle this challenge, an obvious solution is to re-train the model by using newly collected data from the new environment. This is impractical because the changing network does not allow enough time to collect enough new data and then train a new model before violating the latency constraint. Therefore it is a pressing research challenge in multi-antenna communications to achieve fast adaptation of beamforming solutions.

Transfer learning and meta learning have been recognized as two emerging techniques to design adaptive beamforming. Transfer learning [12] has been applied to improve performance of various resource allocation problems in wireless communications due to its strong ability of transferring useful prior knowledge from the old scenario to a new one [13], [14]. In machine learning practice, fine-tuning is a widely used transfer learning technique which re-trains part of a pre-trained model on a new but related task with new data. Different from transfer learning, meta learning is a 'learning to learn' strategy which aims to train a model with better generalization ability [15]. It includes metric-based and optimization-based meta learning. The metric-based meta learning aims to learn to embed input data into a task-specific metric space in which learning of new tasks is efficient. Examples of embedding include Siamese networks, matching networks, relation networks and prototypical networks [16]. The optimization-based meta learning aims to learn the gradient-based neural network optimization such as hyper-parameters so that the neural network can effectively learn new tasks. One of the most popular optimization-based meta learning algorithms is the model-agnostic meta-learning (MAML) algorithm proposed in [17] that is widely applied to general deep neural networks. The MAML algorithm aims to learn a parameter initialization of the deep neural network model, such that a small number of gradient updates of the parameter by using limited testing data from a new task will produce large reduction in the loss function of that task. The MAML algorithm has been successfully used to deal with the end-to-end decoding problem over fading channels [18] [19] with few pilots and to learn the downlink channel state information (CSI) from the uplink CSI in frequency division duplexing (FDD) systems [20]. Our previous work in [21] has also demonstrated that MAML-based adaptive beamforming can rapidly adapt to the new environment and is superior to the transfer learning method.

Although the MAML algorithm has shown a good ability on solving the mismatch issues of beamforming in designing dynamic wireless networks [21], it may cause outdated beamforming prediction due to its high complexity in the training and adaptation processes. Hence, it is important to design a new method, which can reduce the complexity and maintain the performance of adaptive beamforming. The intuition of our proposed approach is based on the findings in [16] and [22] that the dominant factor of the effectiveness of the MAML algorithm is feature reuse rather than the sophisticated design of the meta learning algorithm. Motivated by the results in [16] and [22], the novelty of this paper is to propose a new simple yet effective embedding-based meta learning algorithm to extract the main features and improve the MAML algorithm for adaptive beamforming by reducing the complexity while guaranteeing the efficiency and performance. Our main contributions are summarized as follows:

- We propose a simple yet effective learning framework for adaptive beamforming design, which first trains an embedding model by using existing data in the pre-training stage, and then fits a new model using support vector regression (SVR) with limited new channels and labelled solutions in the adaptation stage. Our proposed method belongs to the metric-based meta learning, so it is different from the transfer learning approach. In addition, different from the existing transfer learning and meta learning methods, our proposed design framework is applicable to both supervised and unsupervised learning in the pre-training stage, and it trains a simple regression model instead of fine-tuning the pre-trained model in the adaptation stage.

- We apply the proposed framework to design specific adaptive beamforming algorithms for the SINR balancing problem and the sum rate maximization problem, which use supervised learning and unsupervised learning, respectively.
- To further investigate the effectiveness of the proposed fast learning framework on beamforming design in non-stationary scenarios, we extend our framework to the online application to solve the beamforming prediction problem in real-time communications systems.
- Extensive simulations are carried out to assess the adaptation capability of the proposed algorithms in realistic communications scenarios. The results verify that the proposed adaptive beamforming algorithms improve the adaptation performance, the stability and the computational efficiency in both pre-training and adaptation stages compared to transfer learning and MAML.

The remainder of this paper is organized as follows. Section II introduces the system model, problem formulation and the existing MAML algorithm. In Section III, the proposed fast meta learning framework and the algorithms for the two applications are presented. Section IV develops the fast online meta-learning based adaptation algorithm. Simulation results and conclusions are presented in Section V and Section VI, respectively.

Notions: All boldface letters indicate vectors (lower case) or matrices (upper case). The superscripts $(\cdot)^H$ and $(\cdot)^{-1}$ denote the conjugate transpose and the inverse of a matrix, respectively. In addition, $||\mathbf{z}||_2$ denotes the L_2 norm of a complex vector $\mathbf{z}$. The operator $\mathcal{CN}(0, \mathbf{\Theta})$ represents a complex Gaussian vector with zero-mean and covariance matrix $\mathbf{\Theta}$. $\mathbf{I}_M$ denotes an identity matrix of size $M \times M$. Finally, $\leftarrow$ denotes the assignment operation.

II. System Model, Problem Formulation and MAML Algorithm

A. System Model and Problem Formulation

We consider a multi-input single-output (MISO) downlink system where a base station (BS) with N_t antennas serves K single-antenna users. The received signal at user k can be written as

$$\mathbf{y}_k = \mathbf{h}_k^H \mathbf{w}_k \mathbf{s}_k + \mathbf{h}_k^H \sum_{j \neq k}^{K} \mathbf{w}_j \mathbf{s}_j + n_k, \tag{1}$$

where $\mathbf{h}_k \in \mathbb{C}^{N_t \times 1}$ denotes the channel vector between the BS and user k, $\mathbf{w}_k$ and $\mathbf{s}_k \sim \mathcal{CN}(0,1)$ denote the transmit beamforming vector and the information-bearing signal for user k with normalized power, respectively. The additive Gaussian white noise (AWGN) is given by $n_k \sim \mathcal{CN}(0, \sigma_k^2)$. As a result, the received SINR at user k is expressed as

$$\gamma_k = \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{j \neq k}^{K} |\mathbf{h}_k^H \mathbf{w}_j|^2 + \sigma_k^2}, \forall k. \tag{2}$$

Based on the aforementioned system setup, we consider a general utility maximization problem under the total power constraint P which is formulated as

$$\max_{\mathbf{w}_k, k=1,\ldots,K} \; U(\gamma_1, \ldots, \gamma_K), \qquad \text{s.t.} \; \sum_{k=1}^{K} \|\mathbf{w}_k\|_2^2 \leq P, \tag{3}$$

where $U(\gamma_1, \ldots, \gamma_K)$ is the utility, which is a non-decreasing function of individual users' SINR. We will consider maximization of the SINR and the sum rate as two examples in Section IV.

Normally the problem (3) is nonconvex and can be solved by using iteration based-optimization algorithms such as [2], [3] but with high computational complexity. Consequently, the computational latency of conventional optimization algorithms will render the beamforming solution obsolete when the CSI changes. Although some existing deep learning-based algorithms [7]–[10] can reduce the complexity, they rely on the assumption that training data and testing data come from the same channel distribution to avoid serious performance deterioration caused by task mismatch issues. This strict assumption is difficult to satisfy in practical scenarios. Fortunately, a common optimization problem with the same objective function and constraints will be solved in different channel conditions, and some inherent features exist in the solution structure of such an optimization problem which will help the beamforming design. Therefore, how to reuse such common features to design a fast meta-learning based beamforming algorithm to cope with the dynamic wireless environments is the main objective of our paper.

B. Overview of MAML

Although the MAML algorithm proposed in [17] has been considered as an efficient optimization-based meta learning method for solving the mismatch issue, the complex optimization procedures also cause a high complexity. As a comparison to our proposed algorithm, we provide a brief overview of the MAML algorithm.

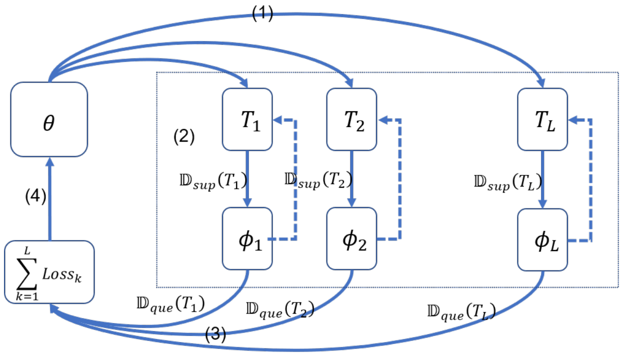


Fig. 1. The workflow of the meta-training stage of the MAML algorithm: Step (1) – network parameter initialization; Step (2) – update of inner loop task parameters; Step (3) – calculation of the loss function of tasks on their query set; Step (4): update of the network parameter.

The MAML algorithm aims to find a good neural network parameter initialization via two optimization loops in the meta training stage as shown in Fig. 1. The inner loop optimization procedure is used to optimize the task-specific parameters for each task based on the initialization of the neural network parameters as shown in step (2) of Fig. 1. The outer loop optimization procedure is used to optimize the initialization of the neural network parameters based on the task-specific parameters as shown in step (3) and (4) of Fig. 1. More specifically, we define the initial neural network parameters as θ, and the training task set as $\{\mathcal{T}(k)\}_{k=1}^{L}$, which includes L tasks. For each task, the inner loop procedure aims to optimize its own task-specific parameters ϕ_k on its support set $\mathbb{D}_{sup}(k)$, which can be expressed as

$$\phi_k = \arg\min_{\phi_k} \mathrm{Loss}_{\mathbb{D}_{sup}(k)}(\phi_k), \forall k, \tag{4}$$

where $\mathrm{Loss}_{\mathbb{D}_{sup}(k)}$ is the loss function of task k. Based on a few steps of gradient descent, the task-specific parameters ϕ_k can be updated by $\phi_k^{(j)} = \phi_k^{(j-1)} - \beta\nabla_{\phi_k^{(j-1)}} \mathrm{Loss}_{\mathbb{D}_{sup}(k)}(\phi_k^{(j-1)})$, where j, β, and ∇ denote the iteration step, the learning rate, and the gradient function, respectively. Note that $\phi_k^{(0)}$ equals to θ. Based on the task-specific parameters of each task, we define the meta-loss as $\text{Loss}_{meta}(\theta) = \sum_{k=1}^{L} \text{Loss}_{\mathbb{D}_{que}(k)}(\phi_k)$, where $\text{Loss}_{\mathbb{D}_{que}(k)}(\phi_k)$ denotes the loss on the query set $\mathbb{D}_{que}(k)$ of task k after the inner loop updates. Then, the initialization of the neural network parameters can be updated by minimizing the meta-loss via the outer loop optimization procedure, which is expressed as

$$\theta = \arg\min_{\theta} \text{Loss}_{meta}(\theta). \tag{5}$$

By using the gradient descent update, parameters θ can be updated by $\theta \leftarrow \theta - \alpha \nabla_{\theta} \text{Loss}_{meta}(\theta)$, where α is learning rate. Notice that the chain rule is included in calculating the parameters θ.

III. Fast Meta Learning Framework and Applications

The complexity associated with the meta training and meta adaptation stages of the MAML algorithm is still high, which will affect how fast it can react to the changing environment. In this section, we aim to design a simple and efficient adaptation framework that is able to achieve comparable performance as MAML. Our design is motivated by the observation in [16] and [22] that a good embedding model that can extract key features is the most important factor to achieve effective adaptation. In the following, we shall present the proposed fast meta learning framework and its two applications to design adaptive downlink beamforming.

A. Design of Fast Meta Learning Framework

The process of the proposed fast adaptation algorithmic framework is illustrated in Fig. 2, where f_θ is the pre-trained embedding model. Since feature reuse is the main reason for MAML to achieve good adaptation performance, it is possible to train a feature extractor without multiple tasks and two optimization loops which are two factors that affect the training efficiency of MAML.

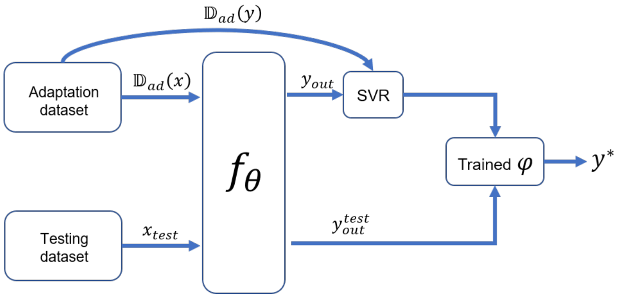


Fig. 2. The workflow of the proposed fast adaptation algorithm.

Rather than designing a new meta learning algorithm, we aim to design a simple way to train an embedding model for feature extraction based on a single task $\mathcal{T}$, which merges all meta training data $\mathcal{D}_{fast} = \cup\{\mathbb{D}_{sup}(k), \mathbb{D}_{que}(k)\}_{k=1}^{L}$. Based on the single task $\mathcal{T}$, the simple embedding model can be obtained by solving the following optimization problem

$$\theta = \arg\min_{\theta} \mathrm{Loss}_{\mathcal{D}_{fast}}(\theta), \tag{6}$$

where $\mathrm{Loss}_{\mathcal{D}_{fast}}(\cdot)$ is the loss function defined as the difference between the predicted output and the target output. The parameter of the embedding model can be updated using the gradient-based method as

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} \mathrm{Loss}_{\mathcal{D}_{fast}}(\theta). \tag{7}$$

Since the process of training the embedding model is similar to the conventional deep neural network training, which does not need to apply the alternating procedures, it is more time efficient compared to the model training of the MAML algorithm and this will be verified numerically. In addition, depending on the specific problem, the training of the embedding model can use other methods such as Siamese networks and matching networks [16], so our proposed embedding model based method is different from and much more general than transfer learning.

As mentioned before, a direct employment of the pre-trained model to achieve the prediction on new tasks causes performance degradation, and fortunately feature reuse from the old tasks to the new tasks is able to avoid such degradation. Hence, we can use the pre-trained embedding model to first extract the features of the new tasks. However, how to make use of the extracted features to achieve fast adaptation is still a challenge. We propose to use SVR, which is a fast regression algorithm based on the support vector machine, as a solution to post-process the extracted features for fast adaptation. The SVR technique aims to find a hyperplane, which has the shortest distance to all data points. Specifically, the features of a new task are extracted by using the pre-trained embedding model over the associated adaptation dataset $\mathbb{D}_{ad}$, which is expressed as

$$y_{out} = f_{\theta}(\mathbb{D}_{ad}(x)), \tag{8}$$

where f_{θ} denotes the pre-trained embedding model with the parameter θ, $\mathbb{D}_{ad}(x)$ and y_{out} denote the input dataset with N_{ad} samples and the output features of the embedding model with dimensions of $N_{ad} \times 2 \times N_t K$ and $N_{ad} \times K$, respectively. $\mathbb{D}_{ad}(y)$ is the final output data in the adaptation dataset with a dimension of $N_{ad} \times K$ associated with $\mathbb{D}_{ad}(x)$. Then y_{out} and the associated labelled data $\mathbb{D}_{ad}(y)$ are used as input and output to train the SVR model to predict the final result. The parameter φ of the SVR model, which includes the weight W and the bias b, can be obtained by minimizing the loss function below (which also includes a nonlinear kernel function), i.e.,

$$\varphi^* = \arg\min_{\varphi} \text{Loss}_{\mathbb{D}_{ad}(y)}(W y_{out} + b, \mathbb{D}_{ad}(y)). \tag{9}$$

Once we have the pre-trained embedding model f_θ and the SVR model denoted by $f(\varphi^*)$, we can use them to find out the adaptive solution in the testing stage. Full details of the proposed fast adaptation solution are summarized in Algorithm 1.

Compared to the MAML algorithm, our proposed fast meta-learning method has two advantages. First, the proposed fast adaptive algorithm has lower complexity in both training and adaptation stages by using a simpler training process and avoiding the fine-turning of the pre-trained model. In the adaptation phase, for our proposed use of SVR model, there are K^2+K parameters to optimize, while the MAML algorithm needs to optimize $FN^2(F+2)+2F+K^2N_tF+K$ variables which are the same as the original neural network, where F is the size of the filter in the CNN module. Furthermore, the transfer learning method has K^2N_tF+K parameters to optimize corresponding to the last layer of the original neural network. Second, our proposed method can be used in either supervised or unsupervised learning in the pre-training stage as long as a good embedding model is obtained. In contrast, the MAML method relies on supervised learning and requires labelled data in the pre-training stage.

Algorithm 1: The proposed fast adaptation framework for a regression problem.

```
Input: Learning rate $\alpha$, meta training dataset $D_{fast}$, adaptation dataset $\mathbb{D}_{ad}$, testing dataset $\mathbb{D}_{test}$
Output: Predict value $y^*$

Embedding model training
Randomly initialize the network parameter $\theta$
while not done do
    $\theta \leftarrow \theta - \alpha \nabla_\theta \text{Loss}_{D_{fast}}(\theta)$ or by using the ADAM optimizer
end while

Adaptation and testing
−−−−−−−−−−−−−−−−−−−−−− Adaptation −−−−−−−−−−−−−−−−−−−−−−
Extract the feature using pre-trained embedding model $f_\theta$ on the adaptation dataset: $y_{out} = f_\theta(\mathbb{D}_{ad}(x))$
Train the SVR model $\varphi$ based on $y_{out}$ and $\mathbb{D}_{ad}(y)$ to obtain $f_{\varphi^*}$
−−−−−−−−−−−−−−−−−−−−−−−Testing −−−−−−−−−−−−−−−−−−−−−−
Extract the feature using pre-trained embedding model $f_\theta$ on the testing dataset: $y_{out}^{test} = f_\theta(\mathbb{D}_{test}(x_{test}))$
Predict the output using the extracted feature: $y^* = f_{\varphi^*}(y_{out}^{test})$.
```

B. Applications

In this subsection, the effectiveness of the proposed fast adaptive method will be verified by using two applications of utility maximization: SINR balancing and sum rate maximization. Its optimal solution of the former problem can be found by using numerical algorithms, whereas it is not easy to find the optimal solution of the latter problem. Some iterative algorithms can be used to find the sub-optimal solution of the sum rate maximization problem, such as the weighted minimum mean squared error (WMMSE) algorithm [4], [24]. Therefore, their training methods for the embedding model are different, but we will show that both can be handled by the proposed framework.

1) ***SINR Balancing****:* Based on the system model in Section II, the SINR balancing problem under the total power constraint P can be formulated as

$$\max_{\mathbf{w}_k, k=1,\ldots,K} \min_{1\leq k\leq K} \gamma_k, \quad \text{s.t.} \sum_{k=1}^{K} \|\mathbf{w}_k\|_2^2 \leq P. \tag{10}$$

Directly predicting beamforming causes high training complexity and inaccurate results due to the high dimensional beamforming matrix. As mentioned before, feature reuse is a key concept to achieve good adaptation performance. Therefore, we propose to predict the low dimensional uplink power allocation vector as the main feature vector based on which the original beamforming matrix can be readily recovered. According to the uplink-downlink duality in [3] and [10], using the uplink power allocation vector to recover beamforming is possible because the same SINR region of the uplink and downlink problems can be achieved given the same total transmit power. Based on uplink-downlink duality and by defining the normalized beamforming $\mathbf{w}_k = \tilde{\mathbf{w}}_k\sqrt{p_k}$, the downlink problem (10) can be converted into the following uplink problem

$$\max_{\mathbf{q}} \min_{1\leq k\leq K} \frac{q_k|\mathbf{h}_k^H\tilde{\mathbf{w}}_k|^2}{\sum_{j\neq k}^{K} q_j|\mathbf{h}_j^H\tilde{\mathbf{w}}_k|^2+\sigma_k^2}, \quad \text{s.t.} \ \|\mathbf{q}\|_1 \leq P, \ \|\tilde{\mathbf{w}}_k\|_2 = 1, \forall k, \tag{11}$$

where $\mathbf{q} = [q_1, \ldots, q_K]^T$ and q_k is the uplink power allocation for user k, $\tilde{\mathbf{w}}_k$ and p_k are the normalized beamforming and downlink power allocation of user k, respectively. With the predicted uplink power allocation vector, the original downlink beamforming can be recovered as follows.

First, the normalized beamforming vector can be obtained as $\tilde{\mathbf{w}}_k = \frac{(\sigma_k^2\mathbf{I}+\sum_{k=1}^{K} q_k\mathbf{h}_k\mathbf{h}_k^H)^{-1}\mathbf{h}_k}{\|(\sigma_k^2\mathbf{I}+\sum_{k=1}^{K} q_k\mathbf{h}_k\mathbf{h}_k^H)^{-1}\mathbf{h}_k\|_2}, \forall k$ and then the optimal downlink power allocation vector $\mathbf{p} = [p_1, \ldots, p_K]^T$ can be obtained by using eigenvalue decomposition as detailed in the duality result in [3]. To be specific, the optimal downlink power allocation vector $\mathbf{p}$ is obtained by finding the first K components of the principal eigenvector of the following matrix

$$\boldsymbol{\Upsilon}(\tilde{\mathbf{W}}, P) = \begin{bmatrix} \mathbf{DU} & \mathbf{D}\sigma \\ \frac{1}{P}\mathbf{1}^T\mathbf{DU} & \frac{1}{P}\mathbf{1}^T\mathbf{D}\sigma \end{bmatrix},$$

where $\mathbf{1} = [1, 1, \ldots, 1]^T$, $\mathbf{D} = \mathrm{diag}\{1/|\tilde{\mathbf{w}}_1^H\mathbf{h}_1|^2, \ldots, 1/|\tilde{\mathbf{w}}_K^H\mathbf{h}_K|^2\}$, $\sigma = [\sigma_1^2, \sigma_2^2, \ldots, \sigma_K^2]^T$, and $[\mathbf{U}]_{kk'} = |\tilde{\mathbf{w}}_{k'}^H\mathbf{h}_k|^2$, if $k' = k$, otherwise $[\mathbf{U}]_{kk'} = 0$. Finally, the downlink beamforming matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_k]$ is derived as $\mathbf{W} = \tilde{\mathbf{W}}\sqrt{\mathbf{P}}$, where $\tilde{\mathbf{W}} = [\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_K]$ and $\mathbf{P} = \mathrm{diag}(\mathbf{p})$. The main advantages of predicting the uplink power allocation vector $\mathbf{q}$ rather than the original beamforming matrix are to improve accuracy and to reduce complexity by reducing the output dimension from $2N_tK$ (the beamforming matrix) to K.

Using the above analysis, we choose the uplink power as the output of the neural network. The sample pairs in the meta training dataset D_{fast} include the input channels and output uplink power allocation vectors, which are generated by solving the problem in (11). The sample pairs in the adaptation and training datasets can be generated in a similar way. In the pre-training stage of Algorithm 1, the embedding model will be learned by using supervised training over D_{fast}, while the adaptation and testing processes will be carried out by using the datasets accordingly based on the description in Algorithm 1.

*2) **Sum Rate Maximization:*** Based on the system model in Section II, the sum rate maximization problem under the total power constraint P can be formulated as

$$\max_{\mathbf{W}} \sum_{k=1}^{K} \log_2(1 + \gamma_k), \quad \text{s.t.} \sum_{k=1}^{K} ||\mathbf{w}_k||_2^2 \leq P. \tag{12}$$

Different from the SINR balancing problem in (10), (12) is a nonconvex and well-known challenging problem and no practical algorithm is available to find the optimal solution in an efficient way, so it is difficult to generate enough labelled data. However, we will still be able to exploit key features of the optimal solution to sum rate maximization to facilitate the adaptive algorithm design.

According to the results in [25], the optimal downlink beamforming vectors for the sum rate maximization problem follows the parameterized structure below

$$\mathbf{w}_k^* = \sqrt{p_k}\frac{(\mathbf{I}_N + \sum_{k=1}^{K}\frac{q_k}{\sigma^2}\mathbf{h}_k\mathbf{h}_k^H)^{-1}\mathbf{h}_k}{||(\mathbf{I}_N + \sum_{k=1}^{K}\frac{q_k}{\sigma^2}\mathbf{h}_k\mathbf{h}_k^H)^{-1}\mathbf{h}_k||_2}, \forall k, \tag{13}$$

where q_k is the virtual uplink power and $\sum_{k=1}^{K} q_k = \sum_{k=1}^{K} p_k = P$. The solution structure in (13) provides the required features $\mathbf{q}$ and $\mathbf{p}$ for the beamforming design in the problem (12). To simplify the design, it is shown in [11] that (13) can be further simplified as

$$\mathbf{w}_k^* = \sqrt{q_k}\frac{(\mathbf{I}_N + \sum_{k=1}^{K} \frac{q_k}{\sigma^2}\mathbf{h}_k\mathbf{h}_k^H)^{-1}\mathbf{h}_k}{||(\mathbf{I}_N + \sum_{k=1}^{K} \frac{q_k}{\sigma^2}\mathbf{h}_k\mathbf{h}_k^H)^{-1}\mathbf{h}_k||_2}, \forall k, \tag{14}$$

with negligible performance loss by using the same uplink and downlink power, i.e., $\mathbf{q} = \mathbf{p}$. As a result, we can still use the uplink power as the main features in the training process as the SINR balancing problem.

Unfortunately, there does not exist a low-complexity algorithm in the literature that can find the optimal p_k^* and q_k^* in (13) or q_k^* in (14). In order to obtain the embedding model in the pre-training stage of Algorithm 1, we train the neural network in an unsupervised learning way without relying on labelled data, whose loss function takes the objective function of sum rate directly as the metric, i.e.,

$$\text{Loss} = -\frac{1}{2KL}\sum_{l=1}^{N}\sum_{k=1}^{L}\log_2\left(1+\gamma_k^{(l)}\right), \tag{15}$$

where L is the batch size.

In the adaptation stage, labelled data are still needed. The WMMSE algorithm is a good candidate to find the locally optimal solutions [4], [24], so we will use the WMMSE algorithm to generate the required labelled data, i.e., the uplink power vector $\mathbf{q}$ for the adaptation stage. Based on the labelled data in the adaptation dataset, the SVR model will be trained by minimizing the following loss function, which uses the MSE metric, i.e.,

$$\text{Loss} = \frac{1}{2LK}\sum_{l=1}^{L}\left(||\underline{\mathbf{q}}^{(l)} - \hat{\mathbf{q}}^{*(l)}||_2^2\right), \tag{16}$$

where $\underline{\mathbf{q}}^{(l)}$ denotes the power vector obtained from the WMMSE algorithm, and $\hat{\mathbf{q}}^{*(l)}$ is the predicted result in the adaptation stage.

Before we conclude this section, we provide details of the neural network architecture to train the embedding model and adaptation model as illustrated in Fig. 3. Because both problems will extract the uplink power as the main feature, we can use a unified neural network architecture for embedding. We adopt the model-based beamforming neural network (BNN) framework proposed in [10] as the network structure to obtain the embedding function $f(\theta)$. It takes the channel as input and the output feature is the uplink power vector $\hat{\mathbf{q}}$. This framework is composed of a convolutional neural network (CNN) architecture followed by the fully connected (FC) layer.

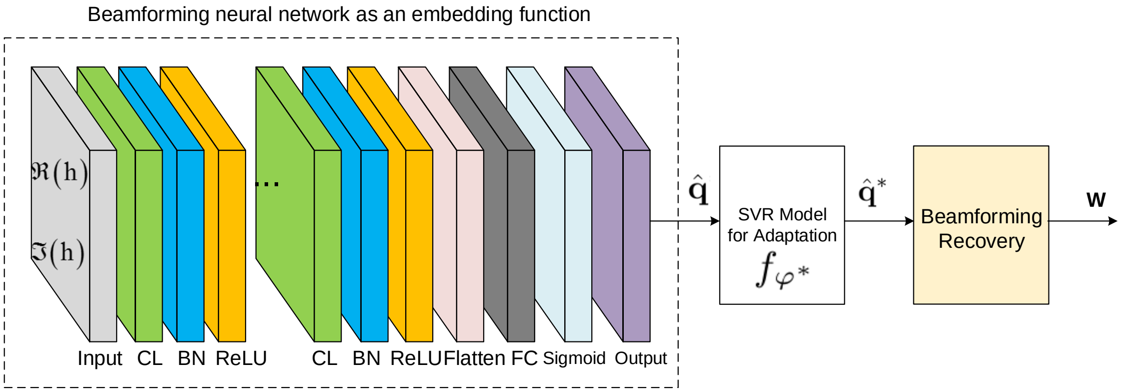


Fig. 3. An illustration of the proposed neural network architecture including the pre-trained embedding BNN model, the SVR model for adaptation and beamforming recovery.

CNN is chosen as the base of the embedding framework due to its ability of extracting features and reducing learned parameters. Specifically, the neural network used for our algorithm includes eleven layers: one input layer, two convolutional layer (CL) layers, two batch normalization (BN) layers, three activation (AC) layers, one flatten layer, one FC layer and one output layer. The complex channel input is split into two real value inputs, so the input dimension of the input layer is $2 \times N_t K$. For the two CL layers, each CL layer applies 8 kernels of size 3×3, one stride, and one padding. The input size of the first CL layer is equal to the size of the input data. The input size of the second CL layer and the output size of both CL layers are equal to $2 \times N_t K \times 8$. Besides, ReLU and Sigmoid functions are adopted at the first two activation layers and the last activation layer, respectively. Adam optimizer is adopted [23] for the optimization of the neural network. In the adaptation stage, the BNN output $\hat{\mathrm{q}}$ is used as the input to train the SVR model, which then outputs the final adaptive uplink power vector $\hat{\mathrm{q}}^*$. The beamforming recovery module is designed based on the uplink-downlink duality introduced in Section III.B-1.

IV. Online Application of Adaptive Algorithms

Although the above proposed algorithm can achieve the fast adaptive beamforming design based on the limited data of the new environment, they work in an offline manner assuming that the data used for adaptation is available in advance and the environment is stationary. They cannot cope with constantly changing situations such as vehicular communications and therefore online algorithms are needed. This is because the channel data is likely to become available only sequentially since effective channel estimation methods take time to first obtain the channel statistics and then estimate the channel, and the channel distribution may be non-stationary as the environment keeps changing. Hence, it is necessary to design an efficient online adaptation algorithm to continuously adapt the model according to the changing environment in real-world applications. In the following, we will explain how to extend the idea of the proposed offline adaptation algorithm to the online scenario. As a comparison to our online adaptation algorithm, we will first provide a brief description about the online MAML learning method.

A. MAML-based Online Learning

The purpose of online machine learning is to achieve continuous learning of prediction using sequential and non-stationary data, and the designed online algorithms should be able to update the best predictor for future data at each time slot. There exist some successful online learning algorithms, such as follow the leader (FTL) [26] and improved FTL [27]. Although FTL is a standard algorithm for online learning, it may not provide the efficient adaptation when the environment changes. The reason is that the principle of training a model using FTL at each time slot is similar to the joint training, which trains the model on a single task. In order to overcome the drawback of FTL when adapting to the new tasks, the authors in [28] incorporate MAML into FTL to design the follow the meta leader (FTML) algorithm.

Similar to the offline MAML algorithm introduced in Section II-B, the FTML algorithm still involves the inner and outer optimization loops in the training stage at each time slot. Since the channel data of each user is sequentially provided, it is impossible to divide tasks in advance like the offline manner. It means that the distribution of tasks is not known in advance. We define $\mathcal{T}_t$ as the task received at time slot t. Since the received task at the current time slot can only be used for adaptation, there is no training process at the first time slot. As the data is accumulated in online learning, we define a task set $\mathcal{B}_t$ to store data of the task t received at the time slot t, $t = 0, \ldots, T$. After the task $\mathcal{T}_t$ is stored at time slot t, the algorithm will sample N_{task} tasks from the previous task sets $\{\mathcal{B}_t,\ t = 0, \ldots, t-1\}$ as the training tasks. The k-th sampled task includes a support set and a query set. Based on the optimization process of inner loop introduced in Section II-B, the task-specific parameters of sampled tasks can be updated via a few step gradient descent. Then, the initialization of the neural network at time slot t can be optimized by using the parameter updating procedure of the outer optimization loop. Once the training process is complete, the task data of the current time slot stored in $\mathcal{B}_t,\ t = 0, \ldots, T$

is used to adapt the trained model. The training and adaptation processes continue to provide online adaptation.

B. The Proposed Fast Online Learning Algorithm

Although the MAML-based online adaptation algorithm can efficiently overcome the mismatch issues in changing environment scenarios, the learning and adaptation processes are sophisticated and time-consuming due to the multiple tasks and two optimization loops. The complex processes may cause delay for real-time wireless communication applications. In order to achieve the rapid adaptation and to satisfy the latency requirements in the real-world communications scenarios, we extend the proposed fast adaptation method presented in Section III to the online scenario to design the fast online adaptive algorithm.

Since we have given the full details of the proposed fast adaptive framework in Section III, this subsection will focus on how to extend it to online applications and especially the adaptation stage. The main idea of the proposed fast adaptation method is to use the pre-trained embedding model to extract common features of the new tasks, and then train a SVR model via the extracted features for adaptation. Before designing its online counterpart, we need firstly to define the datasets. In the online scenario, the data used for adaptation and testing arrives sequentially, which means that the data used in the online learning is accumulated over time. Hence, we define an empty buffer $\mathcal{B}_t$ to store the adaptation data and an empty set D_t to receive the testing data at time slot t. In addition, we assume that the system is able to receive N adaptation sample pairs and U testing sample pairs of each user at each time slot. Note that the adaptation and testing data come from the same distribution. According to different adaptation data used for feature extraction, we propose two online adaptation algorithms: 1) extracting features from adaptation data of the current time slot; 2) extracting features from all adaptation data accumulated until the current time slot. Since the feature extraction and the training of SVR model for both algorithms are the same, we use the first algorithm, which uses the current time slot data for adaptation, to describe our proposed online adaptation algorithm. Specifically, we use time slot t as an example to describe the online adaptation process of the proposed algorithm. At the beginning of time slot t, the algorithm uses buffer $\mathcal{B}_t$ to receive N adaptation sample pairs for each user. Then, based on (8), the extracted feature $\widehat{\mathbf{q}}_t^*$ via the input data in $\mathcal{B}_t$ can be expressed as

$$\widehat{\mathbf{q}}_t^* = f_\theta(\mathcal{B}_t(\mathbf{h}_t)), \tag{17}$$

where f_θ is the pre-trained embedding model, $\mathcal{B}_t(\mathbf{h}_t)$ denotes the input channel stored in the buffer $\mathcal{B}_t$. Next, we will train the SVR model by using the extracted feature $\widehat{\mathbf{q}}_t^*$ and the target output $\mathbf{q}_t$ in $\mathcal{B}_t$. Note that $\mathbf{q}_t$ is the uplink power allocation for both SINR and sum rate maximization problems. Similar to (9), the parameter of the SVR model for our specific regression problems can be expressed as

$$\phi_t = \arg\min_{\phi_t} \text{Loss}(\mathbf{W}_t \widehat{\mathbf{q}}_t^* + \mathbf{b}_t, \mathcal{B}_t(\mathbf{q}_t)), \tag{18}$$

where ϕ_t is the SVR model parameter with the weights $\mathbf{W}_t$ and bias $\mathbf{b}_t$ at time slot t, respectively, and $\mathcal{B}_t(\mathbf{q}_t)$ is the target value stored in $\mathcal{B}_t$ associated with the input $\mathbf{h}_t$. Once the training of the SVR model is completed, we can predict the current testing output based on the trained SVR model and the pre-trained embedding model. Compared to the MAML-based algorithm, the execution time of the proposed fast online adaptive algorithm is significantly reduced because there is no iterative updating procedure in the pre-training stage and adaptation is also simpler, which will be verified by the results in Section V.

As before, the SINR and sum rate maximization problems are considered as two applications to investigate the adaptation ability of the designed fast online algorithm. Both applications take the channel as the input and the uplink power $\mathbf{q}$ as the output. The difference is that the embedding model for the SINR problem is trained by supervised learning while for the sum rate problem, it is trained by unsupervised learning.

V. SIMULATION RESULTS

In this section, numerical simulations are carried out to evaluate the advantages of the proposed adaptive beamforming algorithms for different wireless communications scenarios. We consider a MISO system with one BS and multiple users operating at the carrier frequency of 2.9 GHz, with 20 MHz bandwidth and noise power spectral density of -174 dBm/Hz. Other specific parameters including the number of antennas at the BS N_t, the number of users K and the transmit power P are provided in each figure. We employ the deep learning software Keras with Tensorflow as the backend to pre-train the embedding model and Scikit-learn is used to train the SVR model. We choose the nonlinear Gaussian radial basis function (RBF) kernel in the SVR training. PyTorch is adopted to implement the MAML related algorithms. All simulation results are generated by using a computer with an Intel i7-7700 CPU and a NVIDIA Titan Xp GPU.

In our simulation, the labelled data used for training the embedding model and the SVR model are generated by using three small-scale fading channel models in order to enrich the training dataset: Rayleigh model with distribution $\mathcal{CN}(\mathbf{0}, \mathbf{I}_M)$, Ricean model with the Ricean factor 3, and Nakagami model with a fading parameter 5 and an average power gain 2. For each of the three fading models, we generate 5000 channel samples and obtain 5000 corresponding labelled data for the SINR balancing problem using the algorithm in [3]. Hence, the training dataset includes 15000 sample pairs. For the sum rate maximization problem, we generate 10000 channel samples for each fading model by using the WMMSE algorithm in [4].

We consider the following typical fading scenarios for testing the adaptation capability of the proposed learning algorithms.

- Large-scale fading case: the path loss is given by $\mathrm{PL} = 128.1 + 37.6 \log_{10}(d\ [km])$, where d is the distance between a user and BS. The shadow fading follows the log-normal distribution with zero mean and 8 dB standard deviation. The small-scale fading follows the Rayleigh distribution with zero mean and unit variance.
- WINNER II outdoor case: this is a typical fixed urban scenario specified in WINNER II [29].
- Vehicular case: this is an urban vehicle-to-infrastructure (V2I) scenario defined in Annex A of 3GPP TR 36.885 [30].

For comparison, we introduce other three benchmarks, namely, the optimal solution, the MAML solution, and the non-adaptive learning solution. The definitions of all solutions for comparison are listed below.

- The optimal/suboptimal solution: for the SINR balancing problem, the solution is obtained by using the iterative algorithm proposed in [3]; for the sum rate maximization problem, we consider the WMMSE solution in [4], which is obtained by using an iterative algorithm. It serves as a performance upper bound for all other schemes.
- The MAML solution: this solution shows the adaptation result of the existing MAML algorithm [17] described in Section II-B and Section IV-A for the offline and online scenarios, respectively.
- The transfer learning solution: this solution shows the adaptation result of the existing

transfer learning algorithm [17] that fine tunes the last layer of the neural network to achieve the adaptation.

- The non-adaptive solution: this solution shows the result of using a pre-trained model to predict the beamforming results directly in a different testing environment without any adaptation.

Below we present the results and analysis for the two applications examples.

A. SINR Balancing

We first investigate the large-scale fading case in which the BS is located at the center and all users are randomly distributed within a radius of 500 m. First to decide how many samples will be used in the adaptation, we first study the effect of the number of fine-tuning samples for the same system. Note that more samples will certainly improve the adaptation performance, but also increase the computational complexity and latency. As shown in Fig. 4, SINR increases when the number of fine-tuning samples increases for both MAML and our proposed algorithms and the SINR converges fast for both algorithms. As a result, in the subsequent simulations, we consider 20 adaptation samples as a tradeoff between a low adaptation overhead and satisfactory SINR performance.

In our simulation, we have observed that the performance of MAML and transfer learning solutions could vary depending on the adaptation datasets, so next we study the sensitivity of different algorithms to the random adaptation datasets for an 8-user, 8-antenna system with 25 dBm transmit power in Fig. 5 using 100 random adaptation datasets each with size of 20 samples. We can see that our proposed algorithm achieves robust SINR performance with different adaptation datasets but the achievable SINR of the transfer learning solution varies dramatically and the performance of MAML also fluctuates. In order to obtain stable performance comparison, we average the performance of MAML and transfer learning using 15 random adaptation datasets while for our proposed algorithm, we use just one random adaptation dataset.

We then examine the adaptation performance of the proposed algorithm in Fig. 6. Fig. 6(a) shows the effects of the transmit power on the SINR performance. As expected, the achievable SINR improves as the transmit power increases for all schemes. The SINR performance associated with the proposed algorithm is very close to that of the optimal solution and is better than the MAML solution. The non-adaptive solution achieves the worst SINR performance compared to the adaptive schemes. The performance of the transfer learning solution is only slightly

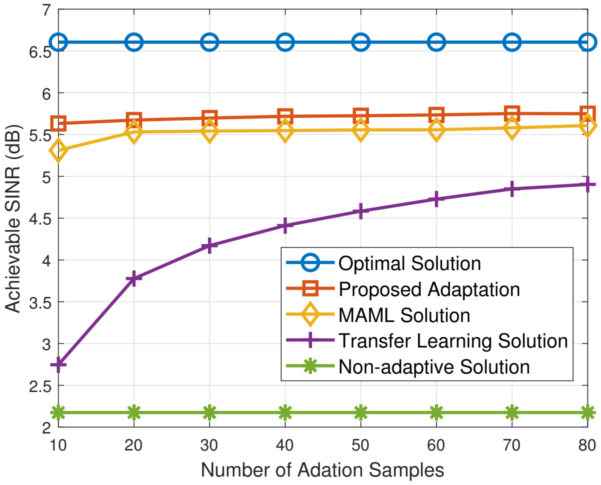


Fig. 4. The effect of the number of fine-tuning samples on the adaptation performance when $N_t = 8, K = 8$, transmit power is 25 dBm.

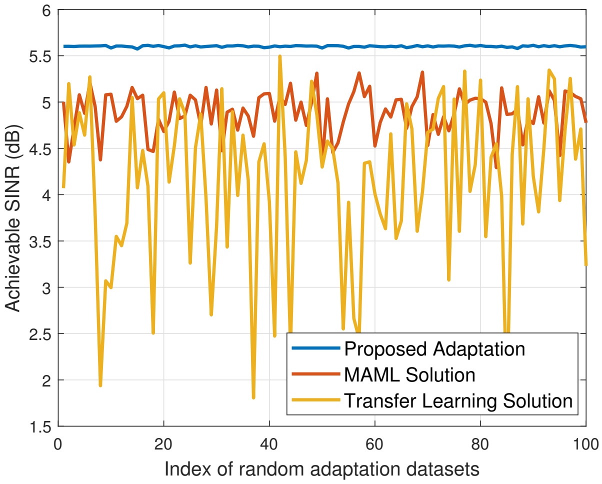
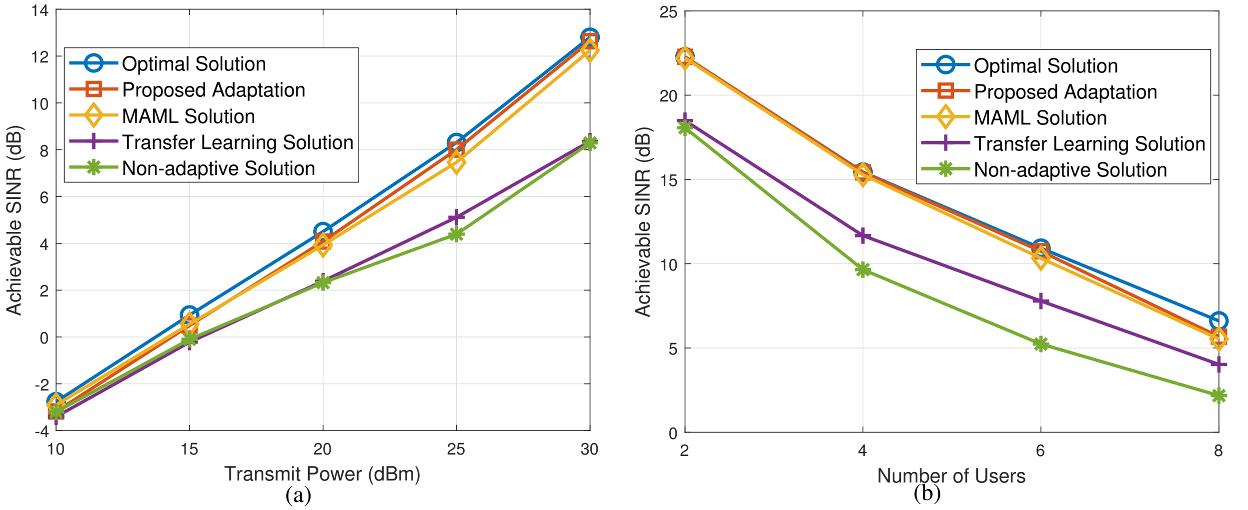


Fig. 5. Sensitivity of the SINR performance versus random adaptation datasets when $N_t = 8, K = 8, P = 25$ dBm.

better than the non-adaptive solution. In Fig. 6(b), the SINR performance versus the number of users is shown. It is observed that as the number of users increases, the performance gap between the MAML algorithm and the optimal solution is enlarged, but our proposed algorithm still achieves better performance. The results in Fig. 6 verify that the proposed fast algorithm provides an efficient adaptive beamforming solution. We compare the execution time of the adaptive algorithms in Fig. 7 versus the number of users. It can be seen that our proposed algorithm achieves more than an order of magnitude gain in terms of both training time and adaptation time compared to the MAML solution and therefore achieves faster adaptation.



(a) (b)

Fig. 6. The SINR performance comparison in the large-scale fading case versus: (a) transmit power when $N_t = 4$, $K = 4$ and (b) the number of users when $N_t = 8$, $P = 25$ dBm.

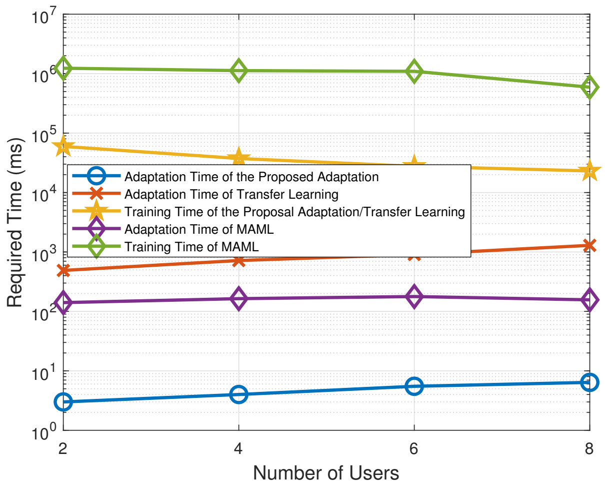


Fig. 7. Comparison of the execution time of the adaptive algorithms.

To get a more accurate comparison of complexity between different algorithms, in Table I, we show the numbers of passes of the neural network. For the pre-training and adaptation stages, each pass includes 100 and 20 channel data, respectively. We have adopted the early stop technique so the number of passes may not change monotonically as the number of users increases. It can be seen that the training of the proposed algorithm requires significantly less passes of the neural network than the MAML algorithm in the pre-training stage. Next we consider the WINNER II outdoor scenario. We assume that the BS is located in the cell center and covers a disc with a radius of 1000 m. Users are randomly distributed between 100 m to 1000 m away

TABLE I

AVERAGE NUMBER OF PASSES OF THE NEURAL NETWORK VS THE NUMBER OF USERS FOR THE SINR BALANCING PROBLEM.

Different Algorithms	**K=2**	**K=4**	**K=6**	**K=8**
Pre-training of MAML	130000	120000	115000	60000
Pre-training of adaptive algorithm/transfer learning	16950	10500	7350	6150
Adaptation of MAML	49	50	51	73
Adaptation of transfer learning	25	47	60	87

from the BS. Fig. 8 demonstrates the adaptation capability of the proposed learning algorithm in the WINNER II outdoor case through the SINR performance. As can be seen from Fig. 8, the proposed algorithm achieves slightly better SINR performance than the MAML solution and both adaptive solutions significantly outperform the non-adaptive solution.

Next, we investigate the performance of the proposed algorithms for V2I case in the urban environment, where we use the Manhattan grid layout with the region size of $750\ m \times 1299\ m$ and grid size of $250\ m \times 433\ m$. There are two 3.5m-wide lanes in each direction; the BS is located in the center of the layout, and the vehicles are uniformly placed on each direction of the road. The probability of each vehicle to change its direction at the intersection is set to 0.4. The velocity of each vehicle is 60 km/h. As seen in Fig. 9, the SINR performance generated by the non-adaptive solution is close to that of adaptive solutions, when the available transmit power budget is low; however, the advantages of the adaptive solution are gradually manifested as the available transmit power budget increases. The possible reason for this interesting observation is that both adaptive and non-adaptive solutions require power to combat the negative effects of the fading channels and there is no enough power left for algorithms to efficiently achieve adaptation at the low power regime. As the transmit power increases, the proposed adaptive algorithm achieves slightly better performance than the MAML solution and much higher SINR performance than the non-adaptive solution.

Next, we evaluate the performance of the proposed online adaptive algorithm in real-world non-stationary scenarios. We investigate the adaptation capability of the proposed online fast adaptation algorithm in changing environments by considering mobile users travelling from outdoor to urban and then highway environments. The freeway case introduced in 3GPP TR 36.885 [30] is used to generate channel data for the highway scenario. The number of lanes

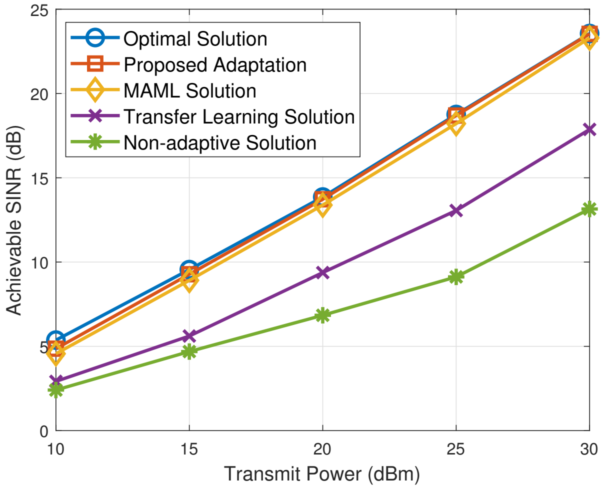


Fig. 8. The SINR performance comparison in the WINNER II outdoor case when $M = 4$, $K = 4$.

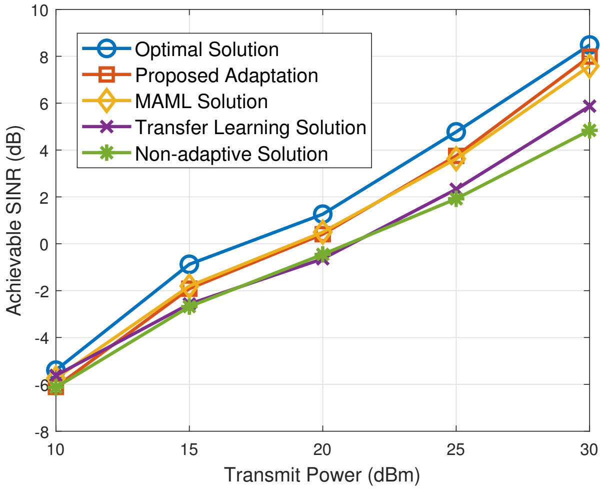


Fig. 9. The SINR performance comparison in the urban case when $N_t = 4$, $K = 4$.

in each direction and the velocity of each vehicle are set as 3 and 120 km/h, respectively. The antenna gains of the BS and vehicles in the urban and highway scenario are set as 8 dBi and 3 dBi, respectively. Fig. 10 shows the adaptation performance comparison between the proposed fast adaptation algorithm and the MAML algorithm over the whole communications period as the users move across different environments. Each simulation point is obtained by averaging all of the actual experimental points at the individual time slots over the previous time slots in the corresponding communication scenario. To implement the simulation, we assume that five adaptation channels $N = 5$ and ten testing channels $U = 10$ of each user are received at

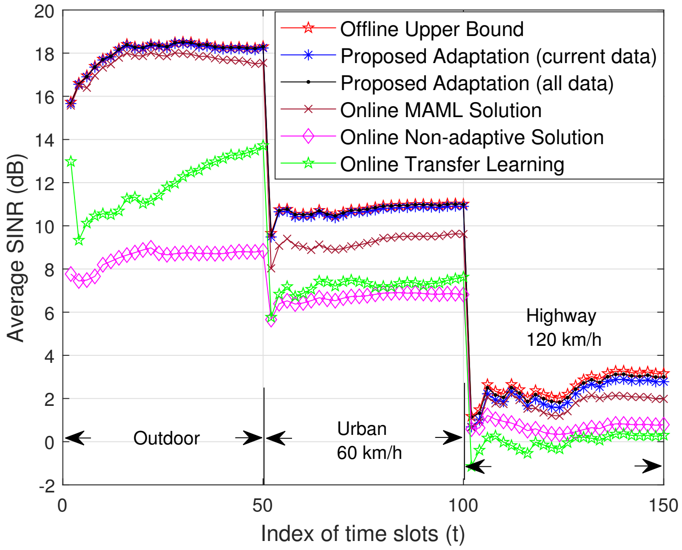


Fig. 10. Performance comparison between the proposed online algorithm and the existing algorithms $M = 4$, $K = 4$, $P = 25$ dBm.

each time slot. Each communications scenario lasts 50 time slots. The offline fast upper bound solution shows the results obtained based on the fast algorithm in the offline manner where the embedding model and the SVR model are trained by using the data of the corresponding communication cases, respectively. The results of two online fast adaptations are obtained by using the data of the current time slot and all available data in the buffer, respectively. As we can see, there exists a significant reduction on the average SINR for all solutions, when the users travel from outdoor to urban and to highway environments due to the changes of fading distribution. This fact indicates that the communication quality in the real-world can be seriously affected by the continuously changing environment. In the figure, the average SINR performance associated with the proposed online fast adaptation algorithms is very close to that of the offline fast upper bound solution, which validates its adaptation ability. In addition, the proposed online fast adaptation algorithms outperform the MAML algorithm and the non-adaptive solution for the whole communications period. It indicates that the proposed algorithm can achieve fast adaptation by efficiently making use of sequential data when the environment changes. It is worth noting that the gap between the online fast algorithms and the MAML algorithm significantly increases when the communications scenario changes from outdoor to urban, whereas the gap reduces when the communications scenario changes from urban to highway. The reason is that there exists obviously difference of channel statistics features between the static scenario (outdoor) and mobility scenario (urban and highway). This fact further verifies that the proposed online fast adaptation algorithm has better adaptation ability when the environment changes abruptly. The online fast adaptation that utilizes all available historical data performs better than that uses only the current data, and this is because more data can train a better SVR model in the adaptation stage. Note that for the urban case and the highway case in the online scenario, the channel coherent times are approximately 6.2 ms and 3.1 ms, respectively. From Fig. 7 and Fig. 10, we can see that our proposed algorithm can finish adaptation within 10 ms or only a few coherent time slots, which confirms that it can adapt to the fast changing wireless environment in real-time.

B. Sum Rate Maximization

In this subsection, we will investigate the performance of the adaptive algorithms to maximize the sum rate. As stated before, there is no practical algorithm that can achieve the optimal beamforming solution for this problem, so we use unsupervised learning in the pre-training stage and use the WMMSE solution to generate labelled data for training of the SVR model in the adaptation stage. To use the MAML algorithm as a benchmark solution, we also use the WMMSE algorithm to generate training data in its pre-training stage. For the non-adaptive solution, we consider both unsupervised training and supervised training using the data generated by the WMMSE solution. In the literature, it has been reported that unsupervised training achieves higher sum rate [10] which will also be verified in our results below.

We first examine the effects of the number of adaptation samples on the adaptive algorithms in Fig. 11 for a 4-user, 4-antenna system. Compared to the results of the SINR balancing problem in Fig. 4, we can see that our proposed algorithm converges more slowly but it is still faster than the MAML algorithm. As a tradeoff between the performance and the complexity, we choose 50 adaptation samples in the subsequent simulations.

We next investigate the sensitivity of the adaptive algorithms to the adaptation datasets in Fig. 12 for an 8-user, 8-antenna system with transmit power of 25 dBm. Fig. 12 shows that there exists a large variance on the sum rate results for both the MAML and the transfer learning solutions when adaptation dataset changes. It indicates that both of these two benchmark solutions heavily depend on the adaptation datasets. By contrast, the proposed adaptive algorithm demonstrates very robust performance when using different adaptation datasets. Similar to the SINR balancing

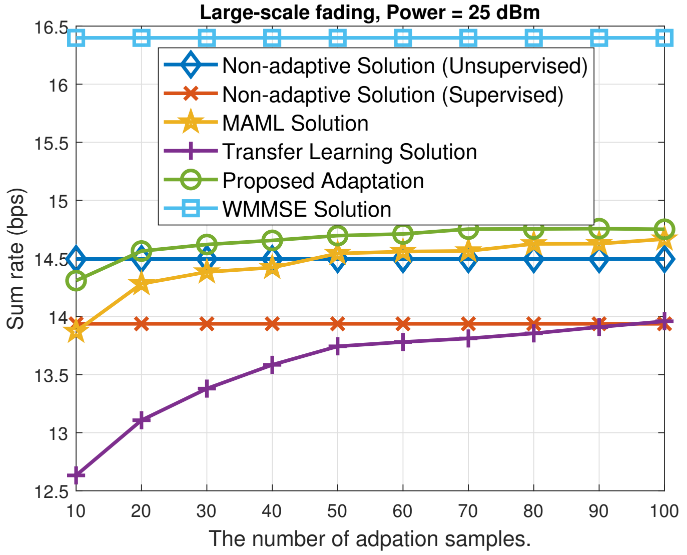


Fig. 11. The comparison of the number of adaptation samples when $M = K = 4$, transmit power is 25 dBm.

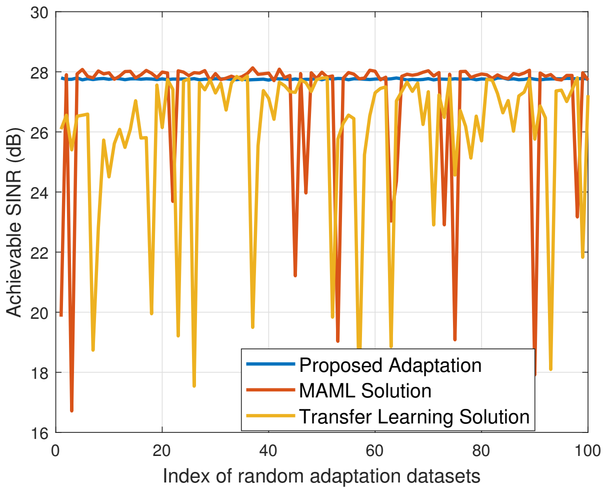


Fig. 12. Sensitivity of the sum rate performance versus random adaptation datasets when $N_t = 8, K = 8, P = 25$ dBm.

problem, we average the results of the MAML and the transfer learning solutions using 15 random adaptation datasets in order to obtain stable comparison results.

The sum rate results in the large scale fading case are presented in Fig. 13. As can be seen in Fig. 13(a), for a 4-user 4-antenna system, our proposed solution achieves slightly better performance than the MAML solution, and the performance gap between different algorithms is not significant. However, Fig. 13(a) shows that as the number of users increases, the performance of transfer learning degrades greatly. The execution time of the adaptive algorithms is compared

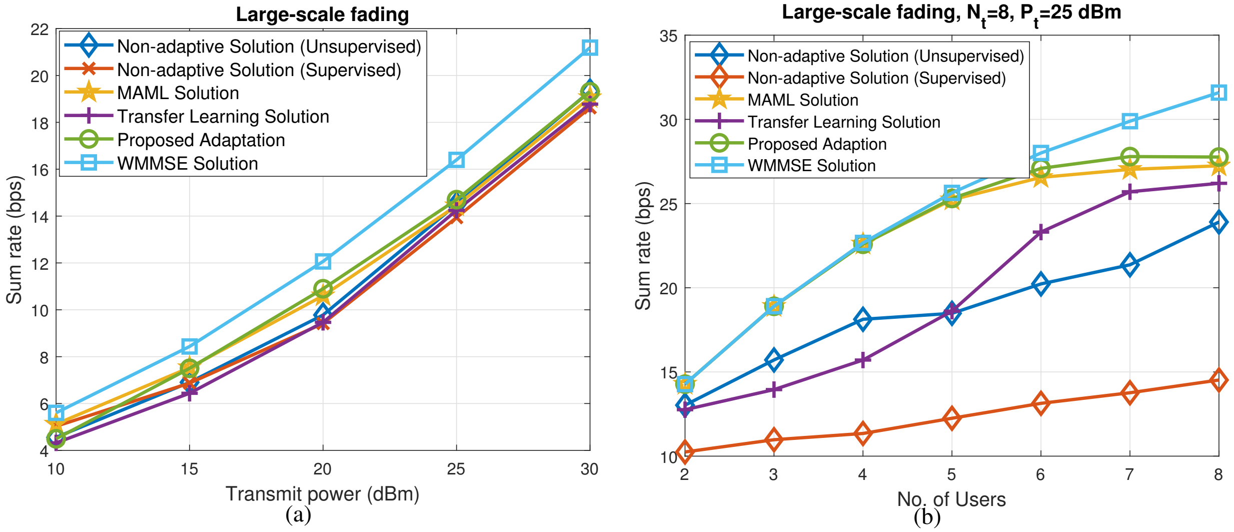


Fig. 13. The sum rate performance comparison on large-scale case for different metrics: (a) transmit power when $N_t = 4$, $K = 4$ and (b) the number of users when $N_t = 8$, $P = 25$ dBm.

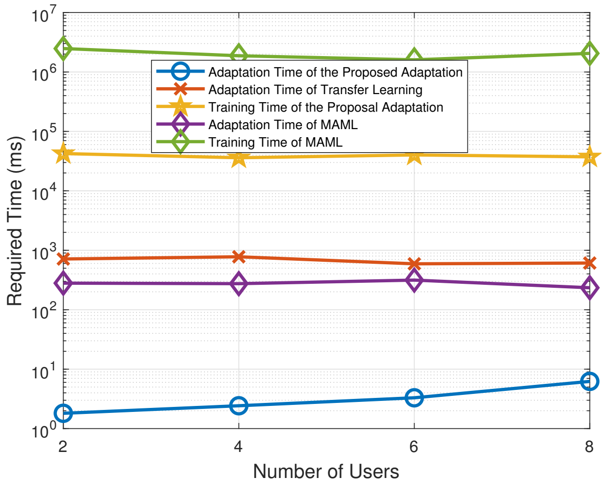


Fig. 14. Comparison of execution time of adaptive algorithms for the sum rate maximization problem when $N_t = 8, P = 25$ dBm.

in Fig. 14 for an 8-antenna system. It can be seen that the proposed algorithm uses less pre-training time than the MAML algorithm. It finishes the adaptation in a few milliseconds and is three orders of magnitude faster than the MAML algorithm. Similar to Table I, we show the numbers of passes of the neural network in Table II. It is observed again that the training of the proposed algorithm requires significantly less passes of the neural network than the MAML algorithm. The transfer learning solution also requires less passes than the MAML algorithm in the adaptation stage.

TABLE II

AVERAGE NUMBER OF PASSES OF THE NEURAL NETWORK VS THE NUMBER OF USERS FOR THE SUM RATE PROBLEM.

Different Algorithms	**K=2**	**K=4**	**K=6**	**K=8**
Training of MAML	255000	185000	155000	200000
Training of adaptive algorithm/transfer learning	7860	6600	7230	6570
Adaptation of MAML	96	102	98	74
Adaptation of transfer learning	52	50	27	21

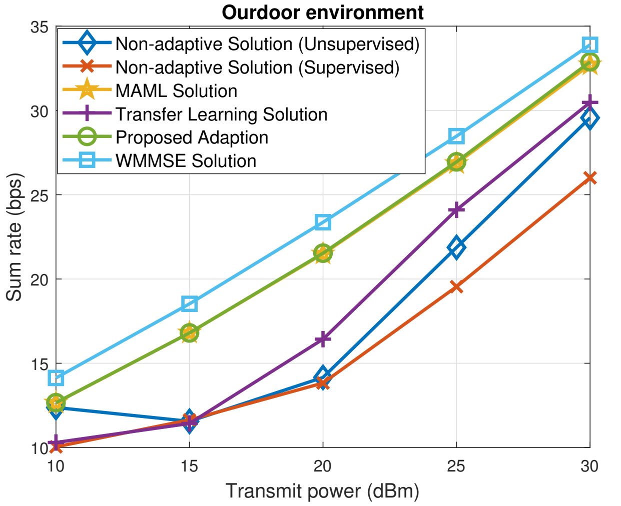


Fig. 15. The sum rate performance comparison in the WINNER II outdoor case when $N_t = 4$, $K = 4$.

The sum rate in the WINNER II outdoor scenario is compared in a 4-user, 4-antenna system. As shown in Fig. 15, both adaptive algorithms achieve good sum rate performance which are close to the results generated by the WMMSE solution. Although the unsupervised learning based non-adaptive solution performs well in the low transmit power regime, its performance degrades fast as the transmit power increases. Similar performance comparison can be observed from the results in Fig. 16 for the V2I case in the urban environment, in which both adaptive solutions achieve the sum rate closer to that of the WMMSE solution than in the outdoor scenario.

Finally we evaluate the sum rate performance of the online algorithms in real-world non-stationary scenarios. Similar to the SINR maximization problem, we consider mobile users travelling from outdoor to urban and then highway environments. Fig. 17 shows the adaptation performance comparison between the proposed fast adaptation algorithm and the MAML algorithm over the whole communications period as the users move across different environments. To

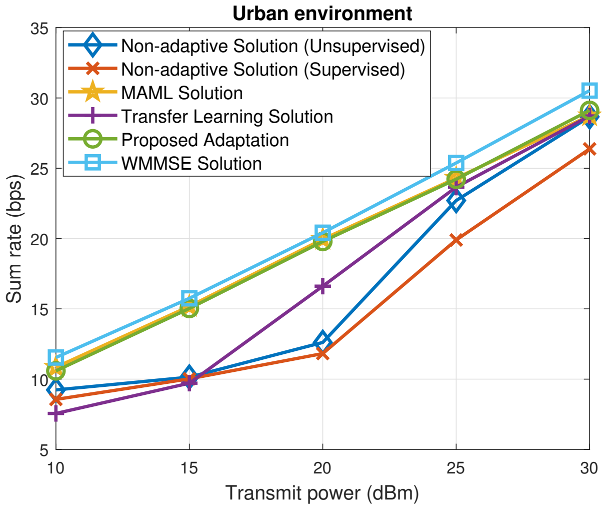


Fig. 16. The sum rate performance comparison in the urban case when $N_t = 4$, $K = 4$.

implement the simulation, we use the same number of adaption channels and testing channels as in the SINR balancing problem. As before, the users' sum rate drops significantly when users cross borders of different environments. In the figure, the sum rate results achieved by the proposed online fast adaptation algorithm are similar to that of the WMMSE solution and outperform the MAML algorithm and the non-adaptive algorithm, which validates its adaptation ability in the changing environments. We note that the gap between the online fast algorithms and the MAML algorithm increases when the communications scenario changes from outdoor to urban, and from urban to highway environments. This observation further confirms that the proposed online fast adaptation algorithm has better and faster adaptation ability when the environment changes.

VI. Conclusions

In this paper, we proposed a simple and efficient adaptive method for beamforming design in dynamic wireless environments. The core idea of this method is to train a good model embedding that extracts key features, followed by fitting a simple SVR model for fast adaptation. We have extended the proposed algorithm to online adaptation that performs well in constantly changing environments. Simulation results demonstrated that the proposed adaptive algorithms achieve a better adaptation performance with a reduced complexity compared to the existing meta learning algorithm. Our results shed some light on designing adaptive learning algorithms from few data for general resource management problems in wireless networks. As to future work, we envisage

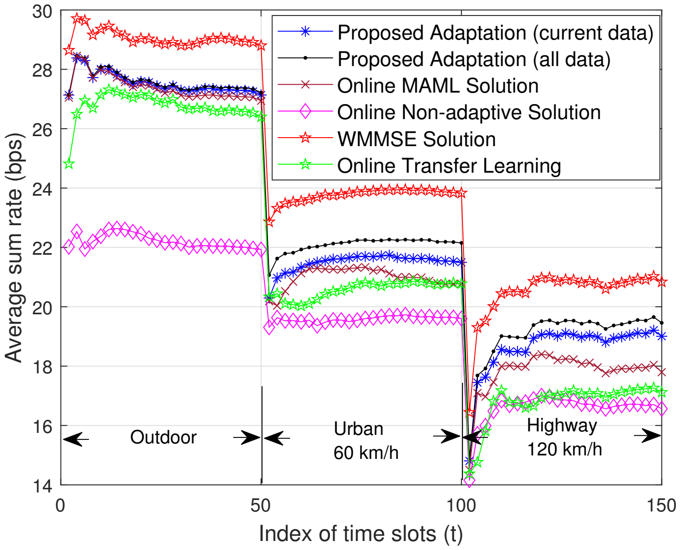


Fig. 17. Sum rate performance comparison between the proposed online algorithm and the existing algorithms, $M = 4$, $K = 4$, $P = 25$ dBm.

that it would be an interesting direction to extend the proposed method to adapt to the varying numbers of active users, and the massive MIMO scenario in which channel estimation and adaptive beamforming will be jointly learned using the deep learning approach.

REFERENCES

[1] F. Rashid-Farrokhi, K. R. Liu, and L. Tassiulas, “Transmit beamforming and power control for cellular wireless systems,” *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1437-1450, Oct. 1998.

[2] M. Bengtsson and B. Ottersten, “Optimal downlink beamforming using semidefinite optimization,” in *Proc. Annual Allerton Conf. on Commun., Control and Computing*, Monticello, NY, Sept. 1999, pp. 987-996.

[3] M. Schubert and H. Boche, “Solution of the multiuser downlink beamforming problem with individual SINR constraints,” *IEEE Trans. Veh. Technol.*, vol. 53, no. 1, pp. 18-28, Jan. 2004.

[4] Q. Shi, M. Razaviyayn, Z. Luo, and C. He, “An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel,” *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331-4340, Sept. 2011.

[5] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu and N. D. Sidiropoulos, “Learning to optimize: training deep neural networks for interference management,” *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438-5453, Oct. 2018.

[6] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, “Deep learning coordinated beamforming for highly-mobile millimeter wave systems,” *IEEE Access*, vol. 6, pp. 37328-37348, 2018.

[7] Y. Shi, A. Konar, N. D. Sidiropoulos, X. Mao, and Y. Liu, “Learning to beamform for minimum outage,” *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5180-5193, Oct. 2018.

[8] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu, “Unsupervised learning based fast beamfrming design for downlink MIMO,” *IEEE Access*, vol. 7, pp. 7599-7605, 2019.

[9] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, “Fast beamforming design via deep learning,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1065-1069, Jan. 2020.

[10] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, “A deep learning framework for optimization of MISO downlink beamforming,” *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1866-1880, Mar. 2020.

[11] J. Kim, H. Lee, S. Hong and S. Park, “Deep learning methods for universal MISO beamforming,” *IEEE Wireless Communications Letters*, vol. 9, no. 11, pp. 1894-1898, Nov. 2020.

[12] 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.

[13] A. Zappone, M. Di Renzo, M. Debbah, T. T. Lam, and X. Qian, “Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks towards wireless systems optimization,” *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 60-69, Jul. 2019.

[14] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, “Transfer learning for mixed-integer resource allocation problems in wireless networks,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1-6.

[15] S. Thrun and L. Pratt, “Learning to learn: Introduction and overview,” in *Learning to Learn*. Springer, 1998, pp. 3-17.

[16] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, “Rethinking few-shot image classification: a good embedding is all you need?” *European Conference on Computer Vision (EECV)*, Glasgow, UK, Aug. 2020.

[17] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proc. 34th Int. Conf. Machine Learning*, 2017, vol. 70, pp. 1126-1135.

[18] S. Park, H. Jang, O. Simeone, and J. Kang, “Learning to demodulate from few pilots via offline and online meta-learning,” *2019, arXiv:1908.09049*, 2019. [Online]. Available: https://arxiv.org/abs/1908.09049

[19] S. Park, O. Simeone, and J. Kang, “Meta-learning to communicate: Fast end-to-end training for fading channels,” *in Proc. IEEE 45th Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Barcelona, Spain, May, 2020, pp. 5075-5079.

[20] 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.*, vol. 68, no. 12, pp. 7485-7497, Dec. 2020.

[21] Y. Yuan, G. Zheng, K. K. Wong, B. Ottersten, and Z.-Q. Luo, “Transfer learning and meta learning ased fast downlink beamforming adaptation, ” *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1742-1755, Mar. 2021.

[22] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, “Rapid learning or feature reuse? towards understanding the effectiveness of MAML,” *International Conference on Learning Representations (ICLR)*, International Conference on Learning Representations (ICLR), Apr.- May 2020.

[23] J. Ba and D. Kingma, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learning Representations (ICLR)*, San Diego, USA, May 2015, pp. 1-15.

[24] S. S. Christensen, R. Agarwal, E. D. Carvalho, and J. M. Cioffi, “Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4792–4799, Dec. 2008.

[25] E. Björnson, M. Bengtsson, and B. Ottersten, “Optimal multiuser transmit beamforming: A difficult problem with a simple solution structure,” *IEEE Signal Process. Mag.*, vol. 31, no. 4, pp. 142–148, Jul. 2014.

[26] J. Hannan, “Approximation to bayes risk in repeated play,” *Contributions to the Theory of Games*, vol. 3, pp. 97-139, 1957.

[27] S. Shalev-Shwartz, “Online learning and online convex optimization,” *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107-194, 2012.

[28] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, “Online meta-learning,” *Proceedings of the 36th International Conference on Machine Learning (PMLR)*, 97:1920-1930, 2019.

[29] P. Kyosti, “IST-4-027756 WINNER II D1.1.2 v.1.2: WINNER II channel models,” *Inf. Soc. Technol.*, 2007.

[30] *3GPP: Technical Specification Group Radio Access Network: Study LTE-Based V2X Services: (Release 14)*, Standard 3GPP TR 36.885 V2.0.0, Jun. 2016