Fast Meta Learning for Adaptive Beamforming

Juping Zhang*, Yi Yuan*, Gan Zheng*, Ioannis Krikidis†, and Kai-Kit Wong‡
*Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, LE11 3TU, UK
†Department of Electrical and Computer Engineering, University of Cyprus, 1678 Nicosia Cyprus
‡Department of Electronic and Electrical Engineering, University College London, London, WC1E 6BT, UK
Email: {j.zhang3, y.yuan, g.zheng}@lboro.ac.uk, krikidis@ucy.ac.cy, kai-kit.wong@ucl.ac.uk

***Abstract*—This paper studies the deep learning based adaptive downlink beamforming solution for the signal-to-interference-plus-noise ratio balancing problem. Adaptive beamforming is an important approach to enhance the performance in dynamic wireless environments in which testing channels have different distributions from training channels. We propose an adaptive method to achieve fast adaptation of beamforming based on the principle of meta learning. Specifically, our method first learns an embedding model by training a deep neural network as a transferable feature extractor. In the adaptation stage, it fits a support vector regression model using the extracted features and testing data of the new environment. Simulation results demonstrate that compared to the state of the art meta learning method, our proposed algorithm reduces the complexities in both training and adaptation processes by more than an order of magnitude, while achieving better adaptation performance.**

***Index Terms*—Meta-learning, embedding model, beamforming.**

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], signal-to-interference-plus-noise ratio (SINR) balancing [2], and sum rate maximization [3]. However, most beamforming solutions are highly complex to implement because of their iterative nature and slow convergence, and therefore cannot meet the critical latency requirement in the fifth generation (5G) and beyond communications systems.

Recently, the deep learning technique has been proposed to address the complexity of beamforming design using the 'learning to optimize' framework [4]. It is based on the intuitive idea that the mapping from the 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. The potential of this approach has been demonstrated in solving a series of beamforming design problems [5]–[8]. However, a major drawback with existing deep learning based beamforming solutions is that they are restricted to a static wireless environment, i.e., the training and testing channel data follow the same distribution, which is unrealistic. In practical wireless networks, the channel distribution may change due to complex environments, high mobility (e.g., in vehicular networks) or unexpected perturbations. Even a well trained model could cause unacceptable performance degradation in the testing environment, which is known as the task mismatch issue. Since re-training a model will violate the latency constraint and is impractical, to tackle this challenge, fast adaptation of beamforming solutions using limited data is urgently needed in multi-antenna communications.

A potential method for adaptation of beamforming is the model-agnostic meta-learning (MAML) algorithm proposed in [9] that is widely applicable 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 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 [10] with few pilots and to learn the downlink channel state information (CSI) from the uplink CSI in frequency division duplexing (FDD) systems [11]. Although the MAML algorithm has a good adaptation ability, it introduces high complexities in the training stage due to iterations used and in the adaptation stage due to fine tuning. To improve the training and adaptation efficiency while maintaining the adaptation performance of MAML, the authors in [12] investigated the key factors that affect the effectiveness of MAML. The results show that feature reuse is the dominant factor for MAML. Based on the results in [12], an efficient learning method was proposed in [13] to solve few-shot image classification by training a linear classifier based on the extracted features rather than using the sophisticated meta training algorithm.

Motivated by the work in [13], this paper aims to design a low-complexity adaptive beamforming solution that can achieve fast adaptation based on limited labelled data in dynamic wireless environments. Our main contribution is that we propose a simple yet effective learning algorithm for adaptive beamforming design. It first learns an embedding model as a feature extractor using existing data in the pre-training stage, and then fits a simple support vector regression (SVR) model based on the extracted features and limited testing data of the new task in the adaptation stage. Compared to MAML, our proposed algorithm simplifies the pre-training design and does not need fine tuning in the adaptation stage. By considering the SINR balancing problem as an example, extensive simulations are carried out to assess the adaptation capability of the proposed algorithm in realistic communications scenarios. The results verify that the proposed adaptive beamforming algorithm outperforms the state of the art MAML algorithm in terms of achieving better adaptation performance and requiring lower computational cost.

II. 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

$$y_k = \mathbf{h}_k^H \mathbf{w}_k s_k + n_k, \tag{1}$$

where $\mathbf{h}_k \in \mathbb{C}^{N_t \times 1}$ denotes the channel vector of user k, $(\cdot)^H$ denotes the conjugate transpose, $\mathbf{w}_k$ and s_k denote the transmit beamforming vector and the information-bearing signal for user k with normalized power, respectively. n_k is the additive Gaussian white noise with zero mean and variance of σ_k^2. The received SINR at user k is expressed as

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

Based on the above model, the SINR balancing problem under the total power constraint P can be formulated as

$$\max_{\mathbf{w}_k, k=1,\ldots,K} \min_{1 \le k \le K} \gamma_k, \quad \text{s.t.} \sum_{k=1}^{K} \|\mathbf{w}_k\|^2 \le P. \tag{3}$$

There exist classical numerical algorithms that can find the optimal solution of problem (3) such as [2], and more recently deep learning based algorithms can solve the problem much faster than the conventional optimization algorithms such as [8]. However, the existing learning methods will cause the task mismatch issue when the testing channel distribution is different from the training channel which happens in dynamic networks. Hence, we focus on the design of fast adaptive beamforming algorithms to overcome the task mismatch issue.

III. Overview of MAML Algorithm

MAML has been recognized as a promising technique to solve the task mismatch issue in beamforming design via training an effective parameter initialization of a deep neural network. In the following, we briefly describe the training and adaptation processes of the MAML algorithm.

A. Definition of Tasks and Datasets

Define a task set $\{\mathcal{T}(k)\}_{k=1}^{L}$, which includes L tasks. Each task is defined as a prediction from a given set of channel realizations. Training data and validation data of each task are randomly selected from the meta-training dataset $\mathbb{D}_{meta}(\cdot)$. We define the set of training data as the support set $\mathbb{D}_{sup}(\cdot)$ and the set of validation data as the query set $\mathbb{D}_{que}(\cdot)$. The support set and the query set of each task include N_s and N_q sample pairs, respectively. The dataset used for adaptation is defined as $\mathbb{D}_{ad}(\cdot)$, which includes N_{ad} sample pairs. Note that the distribution of channel realizations in the adaptation dataset is the same as the distribution of channels in the testing dataset, but different from those in the training dataset.

B. The Meta-training Stage

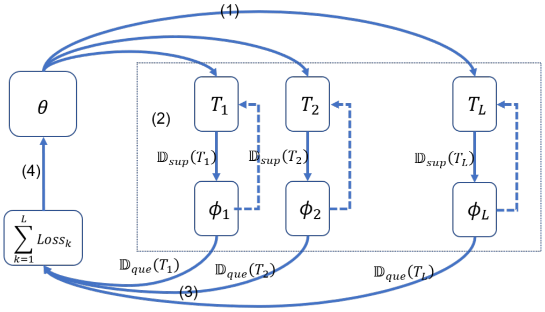


Fig. 1. The workflow of the meta-training stage of the MAML algorithm: Step (1) – network parameter initialization; Step (2) – update of inner loop task parameters; Step (3) – calculation of the loss function of the outer loop task; Step (4): update of the network parameter.

As shown in Fig. 1, MAML uses two optimization loops to obtain the meta initialization in order to achieve fast adaptation on new tasks. Specifically, the inner loop is used to update the parameters of each task based on the associated support set and the global parameter of the network. The outer loop is used to update the initialization of the global neural network parameter based on the trained individual task parameters and the associated query sets. We define a neural network model with the global parameter θ as shown in step (1) of Fig. 1.

In the inner loop, each task aims to optimize its parameter over its support set by using supervised learning as shown in step (2) of Fig. 1, and this can be achieved by

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

where ϕ_k denotes the parameter of task k, ϕ is the generalization notation for the parameter of each task, which is initialized by θ, and $\text{Loss}_{\mathbb{D}_{sup}(k)}$ is the loss function of task k on its corresponding support set $\mathbb{D}_{sup}(k)$. The gradient descent technique can be used to solve this specific problem. In the j-th iteration, the parameter can be updated as

$$\phi_k^{(j)} = \phi_k^{(j-1)} - \beta \nabla_{\phi_k^{(j-1)}} \text{Loss}_{\mathbb{D}_{sup}(k)}(\phi_k^{(j-1)}), \quad \forall k, \tag{5}$$

with

$$\phi_k^{(0)} = \theta - \beta \nabla_{\theta} \text{Loss}_{\mathbb{D}_{sup}(k)}(\theta), \quad \forall k, \tag{6}$$

where β denotes the learning rate.

In the outer loop, the MAML algorithm aims to update the global network parameter θ, which can be obtained by minimizing the sum of the loss functions for all tasks on their corresponding query sets as shown in steps (3) and (4) of Fig. 1, which can be achieved by solving the following problem:

$$\theta = \arg\min_{\theta} \sum_{k=1}^{L} \text{Loss}_{\mathbb{D}_{que}(k)}(\phi_k), \tag{7}$$

where $\mathrm{Loss}_{\mathbb{D}_{que}(k)}$ denotes the loss function of the task k on its corresponding query set $\mathbb{D}_{que}(k)$. The problem (7) can be solved by using the gradient descent technique below:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \sum_{k=1}^{L} \mathrm{Loss}_{\mathbb{D}_{que}(k)}(\phi_k), \tag{8}$$

where α is the learning rate.

C. The Meta-adaptation Stage

In this stage, the pre-trained global network parameter θ is used to adapt the new task over the adaptation dataset $\mathbb{D}_{ad}$. The parameter updating process of the new task is similar to the inner-task loop, which can be expressed as

$$\phi_{ad}^{(j+1)} \leftarrow \phi_{ad}^{(j)} - \beta \nabla_{\phi_{ad}^{(j)}} \mathrm{Loss}_{\mathbb{D}_{ad}}(\phi_{ad}^{(j)}), \tag{9}$$

where ϕ_{ad} is the parameter of the new task and it is initialized by θ, and j denotes the jth iteration. The global network parameter θ has been trained to achieve adaptation, so it is able to adapt to the new task with only a limited amount of data.

IV. The Proposed Fast Adaptation Algorithm

Although MAML has the potential to design adaptive beamforming in dynamic networks, the complexities in the meta training and meta adaptation stages are still high, which will affect how fast it can respond to the changing environment. In this section, we aim to design a simpler and faster adaptation algorithm with comparable performance as MAML. Our design is based on the observation in [12] and [13] that a good embedding model that can extract key features is the most important factor to achieve effective adaptation. In the following, we will present the proposed fast meta learning algorithm and its application to design adaptive beamforming.

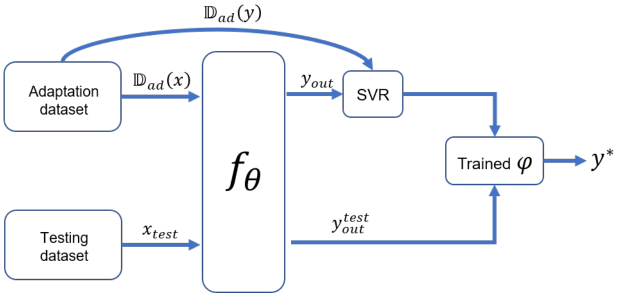


Fig. 2. The workflow of the proposed fast adaptation algorithm.

A. Design of Fast Meta Learning Algorithm

The process of the proposed fast adaptation algorithm is illustrated in Fig. 2, in which f_θ represents the pre-trained embedding model. Feature reuse is the main reason for MAML to achieve good adaptation performance and it does not need multiple tasks and two optimization loops which are two factors that affect the training efficiency of MAML. Therefore we avoid multiple tasks and instead pre-train a simple 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:

$$\theta = \arg\min_\theta \mathrm{Loss}_{\mathcal{D}_{fast}}(\theta), \tag{10}$$

where $\mathrm{Loss}(\cdot)_{\mathcal{D}_{fast}}$ 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

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathrm{Loss}_{\mathcal{D}_{fast}}(\theta). \tag{11}$$

Because the process of training the embedding model is similar to the conventional deep neural network training, which does not include the alternating procedures, it is more time efficient compared to the model training of the MAML algorithm.

As mentioned before, directly using the pre-trained model to achieve the prediction on new tasks causes performance degradation, and feature reuse from the old tasks to the new tasks is the key technique 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. In this paper, we propose to use support vector regression (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. SVR 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

$$y_{out} = f_\theta(\mathbb{D}_{ad}(x)), \tag{12}$$

where y_{out} and $\mathbb{D}_{ad}(x)$ denote the output features and the input of the embedding model, respectively. Then we use the extracted features y_{out} as the input data of the trained SVR model to predict the 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 the nonlinear kernel function chosen):

$$\varphi^* = \arg\min_\varphi \mathrm{Loss}_{\mathbb{D}_{ad}(y)}(W y_{out} + b, \mathbb{D}_{ad}(y)), \tag{13}$$

where $\mathbb{D}_{ad}(y)$ is the target output associated with $\mathbb{D}_{ad}(x)$ in the adaptation dataset.

Once we have the pre-trained embedding model f_θ and the SVR model φ, they can be used 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, the proposed fast adaptive algorithm has lower complexity in both training and adaptation stages by using a simpler training process and avoiding the fine-tuning of the pre-trained model.

```
Algorithm 1: The proposed fast adaptation algorithm
Input: Learning rate $\alpha$, dataset $\mathbb{D}_{fast}$, $\mathbb{D}_{ad}$, and $\mathbb{D}_{test}$
Output: Predicted value $y^*_{test}$
Embedding model training
Randomly initialize the parameter $\theta$
while not done do
    Update $\theta$ according to (11) using the dataset $\mathbb{D}_{fast}$
end while
Adaptation
Feature extraction on adaptation dataset: $y_{out} = f_\theta(\mathbb{D}_{ad}(x))$
Train SVR model $\varphi$ based on $y_{out}$ and $\mathbb{D}_{ad}(y)$
Testing
Feature extraction on testing dataset: $y^{test}_{out} = f_\theta(\mathbb{D}_{test}(x_{test}))$
Predict the testing output: $y^*_{test} = f_\varphi(y^{test}_{out})$
```

B. Selected Feature and Neural Network Structure

Directly predicting beamforming causes high training complexity and inaccurate results due to the high dimensional beamforming matrix. Because feature reuse is key to achieve good adaptation performance, we propose to predict the low dimensional uplink power allocation vector $\mathbf{q}$, as the main feature vector based on which the original beamforming matrix can be readily recovered. According to the uplink-downlink duality in [2] and [8], the same SINR region of the uplink and downlink problems can be achieved with the same total transmit power, so it is possible to use the uplink power allocation vector to recover the downlink beamforming solution by using eigenvalue decomposition as specified in [2]. The main advantage of predicting the uplink power allocation vector $\mathbf{q}$ rather than the original beamforming matrix is to improve accuracy and to reduce complexity by reducing the output dimension from $2N_tK$ (the beamforming matrix) to K.

Before we conclude this section, we will provide details of the neural network employed for pre-training the embedding model. We adopt the model-based beamforming neural network approach proposed in [8] for our algorithm which is composed of a convolutional neural network (CNN) architecture followed by a fully connected (FC) layer. CNN is chosen as the base of our learning framework in this paper 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 $2 \times N_tK$. 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 $2 \times N_tK \times 8$. Besides, ReLU and Sigmoid functions are adopted at the first two activation layers and the last activation layer, respectively. Adam optimizer is adopted [14] for the optimization of the neural network.

V. SIMULATION RESULTS

In this section, numerical simulations are carried out to evaluate the achievable SINR performance of the proposed adaptive beamforming algorithm for different wireless communications scenarios. A MISO downlink system in which one BS with four transmit antennas serves four users is considered, i.e., $N_t = K = 4$, unless otherwise specified. The main simulation parameters are set as follows: carrier frequency is 2.9 GHz, bandwidth is 20 MHz and noise power spectral density is -174 dBm/Hz. We use Tensorflow 2.0.0 and Keras 2.3.1 to train the embedding model and Scikit-learn is used to train the SVR model. PyTorch 1.4.0 is used to implement the MAML algorithm. All simulation results are generated by using a computer with a Intel i7-7700 CPU and a NVIDIA Titan Xp GPU.

In our simulations, the labelled data used for training the embedding model and SVR are generated by using the optimal algorithm in [2] to solve the SINR balancing problem (3). In order to enrich the feature extraction ability of the embedding model, the training data includes three types of small scale fading channel models: Rayleigh model with zero mean and unit variance, Ricean model with the Ricean factor 3, and Nakagami model with the fading parameter 5 and the average power gain 2. For each fading model, we generate 5000 channel samples and obtain 5000 corresponding uplink power values. Hence, the training dataset includes 15000 sample pairs.

We consider the following typical fading scenarios for testing the adaptation capability of the proposed algorithm:

- Large-scale fading case: in this scenario, the path loss is given by $\text{PL} = 128.1 + 37.6\log_{10}(d\ [km])$, 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 the WINNER II channel model [15].
- Vehicular case: this is an urban vehicle-to-infrastructure (V2I) scenario defined in Annex A of the 3GPP release 14 [16].

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

- The optimal solution: this is the solution obtained by using the iterative algorithm proposed in [2]. 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 [9] described in Section III.
- The non-adaptive solution: this solution shows the result of using a pre-trained model to predict the beamforming results in a different testing environment without adaptation. It provides a performance lower bound.

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. To decide how many samples will be used in the adaptation, we first study the effect of the number of adaptation samples for an 8-user 8-antenna system with 25 dBm transmit power. Note that more samples will certainly improve the adaptation performance, but also increase the computational complexity and latency. As shown in Fig. 3, the SINR increases when the number of adaptation samples increases for both MAML and our proposed algorithm and the SINR performance converges fast for both algorithms. As a result, in the subsequent simulations we choose to use 20 adaptation samples as a tradeoff between the low adaptation overhead and the good SINR performance.

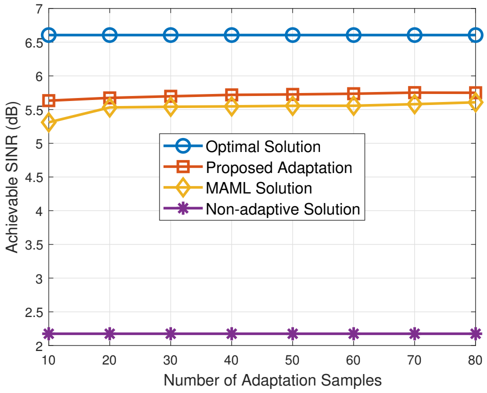
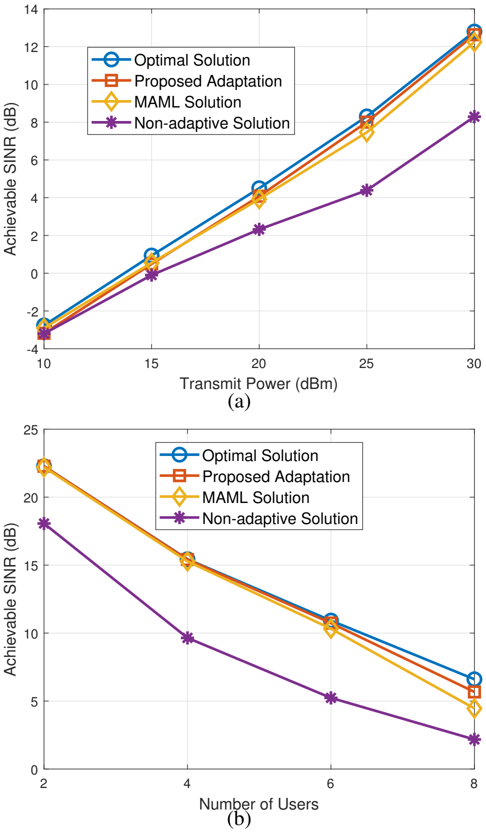


Fig. 3. The effect of the number of adaptation samples on the adaptation performance when $N_t = 8, K = 8$, transmit power is 25 dBm.

Next we examine the adaptation performance of the proposed algorithm in Fig. 4. Fig. 4(a) shows the effect of the transmit power on the SINR performance. As expected, the SINR improves as the transmit power increases for all schemes. The SINR result generated by 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 compared to the adaptive schemes. In Fig. 4(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 adaptive algorithms and the optimal solution is enlarged, but our proposed algorithm still achieves better performance. We compare the execution time of the adaptive algorithms in Fig. 5 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. The results in Fig. 4 and Fig. 5 verify that the proposed fast algorithm provides a better and faster adaptive beamforming solution than the MAML algorithm.



(a)

(b)

Fig. 4. The SINR 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.

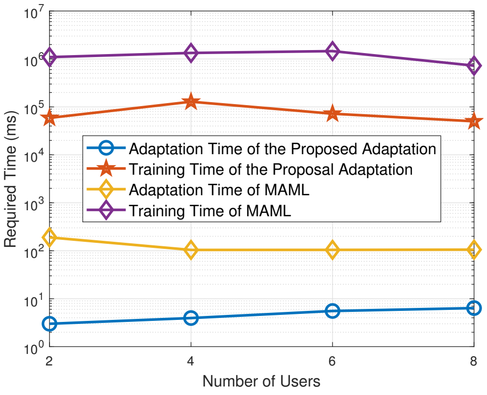


Fig. 5. Comparison of the execution time of the adaptive algorithms when $N_t = 8$, $P = 25$ dBm.

Then, we consider to test the adaptive algorithms in the WINNER II outdoor scenario in which the BS is located in the center of a disc with a radius of 1000 m. Users are randomly distributed between 100 m to 1000 m away from the BS. Fig. 6 demonstrates the adaptation capability of the proposed learning algorithm in the WINNER II outdoor case through the SINR performance. As can be seen, the adaptive solutions significantly outperform the non-adaptive solution, while the proposed adaptive solution achieves slightly better SINR performance than the MAML solution. For the urban V2I case, 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. The vehicles are uniformly placed on

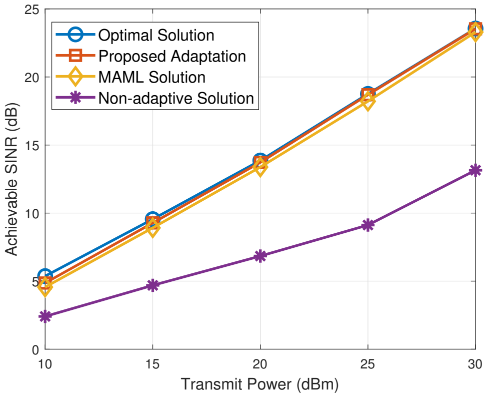


Fig. 6. The SINR performance comparison in the WINNER II outdoor case.

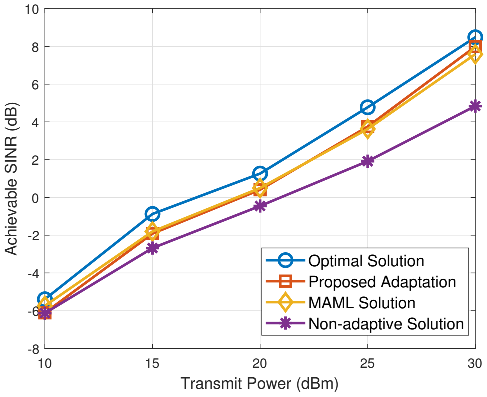


Fig. 7. The SINR performance comparison in the urban V2I case.

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. 7, when the transmit power is low, all schemes have similar and low SINR performance, and this is because they all need to combat the adversarial channel conditions in this scenario. As the transmit power increases, similar to the outdoor case, the proposed algorithm achieves slightly better performance than MAML and much higher SINR than the non-adaptive solution.

VI. Conclusions

In this paper, we proposed a simple and fast adaptation optimization method of beamforming design in dynamic wireless environments. The core idea of the proposed scheme is to learn a good embedding model that extracts key features, followed by training a simple SVR for fast adaptation. Simulation results demonstrated that the proposed adaptive algorithm achieves better performance with lower complexity, compared to the existing meta learning algorithm. We envisage that the proposed adaptive beamforming design algorithm can be extended to many other scenarios such as reconfigurable intelligent surfaces for energy efficiency in wireless communications. Our results also shed some light on designing adaptive learning algorithms from few data for general resource management problems in wireless networks.

Acknowledgment

This work was supported in part by the Leverhulme Trust Research Project Grant RPG-2017-129, the UK Engineering and Physical Sciences Research Council (EPSRC) projects EP/N007840/1 and EP/T015985/1, the European Regional Development Fund and the Republic of Cyprus through the Research and Innovation Foundation, under the projects INFRASTRUCTURES/1216/0017 (IRIDA) and EXCELLENCE/0918/0377 (PRIME).

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] Y. Yang, F. Gao, Z. Zhong, B. Ai, and A. Alkhateeb, "Deep transfer learning based downlink channel prediction for FDD massive MIMO systems," 2019, *IEEE Trans. Commun.*, to appear.

[12] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals, "Rapid learning or feature reuse? towards understanding the effectiveness of MAML," 2019, *International Conference on Learning Representations (ICLR)*, Apr.-May 2020.

[13] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?" 2020, *arXiv:2003.11539*. [Online]. Available: https://arxiv.org/abs/2003.11539

[14] 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.

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

[16] *3GPP: Technical Specification Group Radio Access Network: Study LTE-Based V2X Services: (Release 14)*, Standard 3GPP TR 36.885 V2.0.0, Jun. 2016