

Parameter-Efficient Online Fine-Tuning of ML-Based Hybrid Beamforming with LoRA

Faramarz Jabbarvaziri and Lutz Lampe, *Senior Member, IEEE*

Abstract—We propose low-rank adaptation (LoRA) for machine learning-aided hybrid beamforming (HBF) in episodically dynamic millimeter-wave multiple-input multiple-output (MIMO) systems. This approach introduces low-rank trainable matrices and uses a small buffer with recent channel samples, making it ideal for real-time adjustments. Evaluated for a large MIMO HBF system across both an environment-specific channel using ray tracing and clustered delay line channel models, simulation results show that rank-2 LoRA achieves efficient retraining with only 6% of the original network’s parameters and 128 samples, improving average achievable information rate (AIR) by over 45% compared to the pre-trained model in both scenarios. The method significantly outperforms transfer-learning with full-model online fine-tuning and model-agnostic meta-learning with its “almost-no-inner-loop” variant.

Index Terms—hybrid beamforming, machine learning, low-rank adaptation (LoRA), model-agnostic meta-learning (MAML)

I. INTRODUCTION

Millimeter-wave (mmWave) multiple-input and multiple-output (MIMO) transceivers are an integral component of advanced wireless communication systems. They often incorporate hybrid beamforming (HBF) architectures to balance performance with device cost and power efficiency [1]. Recent work has explored HBF optimization using machine learning (ML) techniques, notably through residual convolutional neural networks (ResNets) with the achievable information rate (AIR) as the loss function to train the ML model [2]–[5]. Learned HBF transfers time-intensive non-convex optimization tasks to the offline training phase, provides improved performance especially when channel models or channel state information (CSI) are imperfect, and benefits from highly parallelizable computations on ML accelerators.

In practical scenarios, channel conditions can vary episodically due to factors such as user mobility and environmental changes [6]. For example, a vehicle or drone with beamforming capabilities might encounter various propagation environments along its path, transitioning from dense urban areas with high multipath effects to more open or suburban settings. Training an ML model on mixed data from multiple scenarios can degrade performance in specific environments, limiting the model’s ability to generalize effectively. Consequently, achieving optimal performance requires periodic or performance-driven online fine-tuning for ML-based beamforming and CSI prediction and compression [6]–[10].

Online fine-tuning for ML-based beamforming has traditionally relied on methods like model-agnostic meta-learning (MAML) [7], [8], [11] and direct transfer learning (TL) with online retraining [7], [10]. Direct TL with online training requires a relatively large number of retraining iterations, making it unsuitable for low-complexity online adaptation in ML-based HBF. Additionally, both MAML and TL approaches also share a key limitation: they require resource-intensive full-model fine-tuning (FMF). To address this, [8] introduced a modular online fine-tuning approach using a MAML variant called almost-no-inner-loop (ANIL) [12] for ML-based HBF. ANIL reduces the number of trainable parameters and data requirements for online training. However, both MAML and ANIL face challenges in the meta-learning stage, as they require second-order derivative (Hessian) calculations for each task in every training iteration. The Hessian matrix in the original MAML method grows quadratically with the number of parameters, making computation and memory costs prohibitively high for large models. This limitation could be mitigated by first-order approximations for Hessian-free MAML [13].

Considering the state-of-the-art ResNet-based HBF techniques [2]–[5] and building on recent advances in low-complexity fine-tuning methods in natural language processing, this paper applies low-rank adaptation (LoRA) [14] to enable rapid adaptation in ML-based beamforming within episodically dynamic channels. LoRA is a parameter-efficient fine-tuning approach that bypasses the computational overhead of offline meta-learning by relying directly on the pre-trained model. This is unlike MAML, which requires a modified initial training process. During online fine-tuning, LoRA only updates low-rank matrices added to the pre-trained weights, significantly reducing the number of trainable parameters and the amount of retraining data needed for effective adaptation. This makes LoRA well-suited for online fine-tuning of ML-based HBF, ensuring computational efficiency and sample effectiveness during channel transitions. Simulation results across two dynamic channel environments—a ray tracing (RT) model and a mix of clustered delay line (CDL) models—show that our LoRA-based method with rank 2 and 128 samples significantly outperforms the pre-trained model, MAML, ANIL, and TL with FMF while maintaining a reduced catastrophic forgetting.

II. SYSTEM MODEL

We consider the downlink of a single-user MIMO orthogonal frequency-division multiplexing (OFDM) HBF system.

We assume that the base station (BS) is equipped with N_B^a antennas and N_B^{RF} radio frequency (RF) chains, while each user device has N_U^a antennas and N_U^{RF} RF chains, operating over K subcarriers. Digital precoding and combining are conducted in the discrete Fourier transform (DFT) domain, with analog precoding and combining applied to the up-converted signal via a network of tunable phase shifters and signal adders.

A. Channel Model

For performance evaluations, we use both the RT simulator from [15] and a statistical channel model. The RT simulator models radio propagation with customizable scenes and propagation paths, which are transformed into channel impulse responses for link-level simulations. For the statistical channel model, we adopt the CDL model, commonly used for simulating scenarios across 0.5 GHz to 100 GHz [16]. The CDL model includes various types (A through E) that cover a wide range of environments a high-mobility user might encounter in an episodically dynamic setting. CDL channels consist of a line-of-sight (LOS) component and N_C non-line-of-sight (NLOS) scattering clusters, each containing N_L scatterers. The $N_U^a \times N_B^a$ channel matrix experienced by the k^{th} OFDM subcarrier for downlink transmission can be written as

$$\mathbf{H}[k] = \sum_{c=0}^{N_C} \sum_{l=1}^{N_L} \alpha_{c,l} \mathbf{a}_r(\beta_{c,l}^r) (\mathbf{a}_t(\beta_{c,l}^t))^H e^{-j2\pi\eta_c \frac{k}{K}} \quad (1)$$

where \mathbf{a}_r and \mathbf{a}_t are the uniform linear antenna (ULA) array responses of the user receiver and the BS transmitter, respectively. Furthermore, $\alpha_{c,l}$ is the path gain, $\beta_{c,l}^r$ and $\beta_{c,l}^t$ are the angle of arrival (AoA) and angle of departure (AoD) of the l^{th} reflecting element of the c^{th} cluster, respectively, and η_c denotes the time lag of the c^{th} cluster. The cluster $c = 0$ corresponds to the LOS component and the path gains account for the Ricean factor.

B. MIMO-OFDM HBF

Let $\mathbf{S}[k] \in \mathcal{A}^N$ denote the signal sent from the BS to the user over subcarrier k of an OFDM symbol, where \mathcal{A} is the quadrature amplitude modulation (QAM) signal constellation of size $M = |\mathcal{A}|$, and N is the number of data streams. Furthermore, let $\mathbf{V}_D[k] \in \mathbb{C}^{N_B^{\text{RF}} \times N}$ and \mathbf{V}_{RF} be the digital and analog precoders and $\mathbf{W}_D[k] \in \mathbb{C}^{N_U^{\text{RF}} \times N}$ and \mathbf{W}_{RF} be the digital and analog combiners, respectively. Moreover, \mathbf{I}_D represents the $D \times D$ identity matrix and $\mathbb{E}(\cdot)$ signifies the expected value operator. Then, we can express the received vector $\mathbf{Y}_{n_s}[k] \in \mathbb{C}^N$ as

$$\mathbf{Y}[k] = \mathbf{G}[k]\mathbf{S}[k] + (\mathbf{W}_{\text{RF}}\mathbf{W}_D[k])^H \mathbf{Z}[k], \quad (2)$$

where

$$\mathbf{G}[k] = (\mathbf{W}_{\text{RF}}\mathbf{W}_D[k])^H \mathbf{H}[k] \mathbf{V}_{\text{RF}}\mathbf{V}_D[k] \quad (3)$$

represents the transmission path gain of the user's data signal. In (2), $\mathbf{Z}[k] \in \mathbb{C}^{N_U}$ denotes additive white Gaussian noise (AWGN) at the user side. We further assume that $\mathbb{E}\{\mathbf{S}[k]\} = \mathbb{E}\{\mathbf{Z}[k]\} = 0$ and $\mathbb{E}\{\mathbf{S}[k](\mathbf{S}[k])^H\} = \mathbf{I}_N$.

During data transmission, $\mathbf{Y}[k]$ is processed to compute log-likelihood ratios (LLRs) for the bits represented by $\mathbf{S}[k]$. For this, we adopt a pragmatic approach as in [2] and separate the data streams using a linear minimum mean-squared error (LMMSE) equalizer. Following [17], during equalization, we assume that the interference plus noise has a known covariance matrix \mathbf{C} . The LMMSE equalizer output signal is then obtained as [2]

$$\mathbf{R}[k] = \text{diag}\left((\mathbf{G}[k])^H(\mathbf{G}[k](\mathbf{G}[k])^H + \mathbf{C})^{-1}\mathbf{G}[k]\right)^{-1} \times (\mathbf{G}[k])^H(\mathbf{G}[k](\mathbf{G}[k])^H + \mathbf{C})^{-1}\mathbf{Y}[k], \quad (4)$$

where $\text{diag}(\cdot)$ returns a square matrix in which the diagonal elements of the input matrix are placed on the main diagonal, and all off-diagonal elements are zero. The LLR for the m^{th} bit associated with the n^{th} symbol in $\mathbf{S}[k]$ is calculated as

$$L^{n,m}[k] = \log\left(\frac{\sum_{x \in \mathcal{A}_{m,1}} e^{-|r^n[k]-x|^2/\sigma^2[k]}}{\sum_{x \in \mathcal{A}_{m,0}} e^{-|r^n[k]-x|^2/\sigma^2[k]}}\right), \quad (5)$$

where $\mathcal{A}_{m,b}$ represents the subset of constellation points with the m^{th} bit label equal to b and $r^n[k]$ denotes the n^{th} element of $\mathbf{R}[k]$. Furthermore, $\sigma^2[k]$ represents the post-equalization noise power of the LMMSE equalizer on the k^{th} subcarrier, which is given in [17, Lemma B.19]. The LLRs obtained from the demodulation in (5) can be used to define a posterior distribution for the associated bit as

$$P(b^{n,m}[k]|r^n[k]) = \frac{1}{1 + e^{(-1)^{b^{n,m}[k]} L^{n,m}[k]}}. \quad (6)$$

Following [18], we can compute the empirical binary cross entropy (BCE)

$$E[k] = -\frac{1}{N} \sum_{m=1}^{\log_2 M} \sum_{n=1}^N \log_2(P(b^{n,m}[k]|r^n[k])) \quad (7)$$

for subcarrier k and average it over several frames with independent channel realizations to obtain $\bar{E}[k]$. The empirical approximation of the AIR is

$$R[k] = \log_2(M) - \bar{E}[k]. \quad (8)$$

III. PROPOSED METHOD

We adopt a DNN structure grounded in established hybrid beamforming literature [4], [5], [19] to optimize the precoding and combining matrices, and we propose a LoRA-based online fine-tuning procedure that activates periodically. The following sections detail the adopted DNN architecture, as well as the training and fine-tuning procedures.

A. DNN Architecture and Training Procedure

To address the requirements for subcarrier-specific and aggregate OFDM signal processing in the digital and analog components of HBF systems, we adopt the model structure proposed in [19], which features two distinct branches for generating the digital and analog parts of the beamforming. In this setup, the analog branch processes channel matrix products summed across all subcarriers, i.e., $\mathbf{P} = \sum_k (\mathbf{H}[k])^H \mathbf{H}[k]$

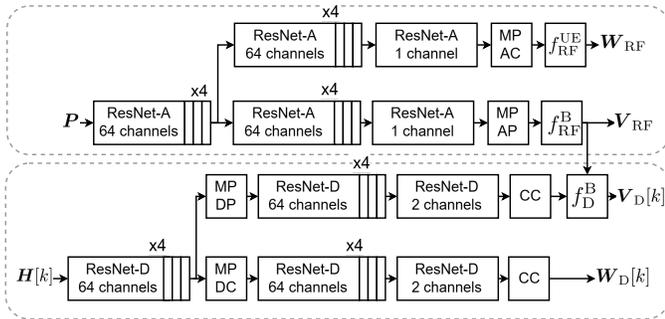


Figure 1. The structure of the considered neural network, includes ResNet-A and ResNet-D modules, which utilize 2D kernels of size 3×3 and 3D kernels of size $3 \times 3 \times 3$, respectively. Max pooling layers (MP), labeled as MP analog combining (MP-AC), MP analog precoding (MP-AP), MP digital combining (MP-DC), and MP digital precoding (MP-DP), have pool sizes of $[1, N_B^a]$, $[N_U^a, 1]$, $[1, \frac{N_U^a}{N_{RF}^a}, \frac{N_B^a}{N}]$, and $[1, \frac{N_U^a}{N_{RF}^a}, \frac{N_B^a}{N}]$, respectively, with padding. These layers ensure that the shapes of data streams from the input channel matrices align correctly with the output precoding and combining matrices. A complex concatenation (CC) layer then combines the two input channels as real and imaginary parts to create a complex-valued tensor. The activation functions f_{RF}^{BS} and f_{RF}^{UE} enforce unit-amplitude constraints on the elements of the analog beamforming matrices, while f_D^B normalizes transmit power as specified in equation (9).

for optimizing the analog precoder and combiners¹, while the digital branch operates on per-subcarrier channels $\mathbf{H}[k]$ generating subcarrier-specific precoding and combining matrices. To reduce the computational cost of the DNN model from [19], we incorporate the ResNet architecture proposed in [4], [5] for HBF optimization.

Figure 1 shows the corresponding architecture of the considered DNN detailing trainable modules and activation functions. To satisfy the unit amplitude constraint for the elements in the analog precoding and combining matrices, Euler's formula is applied [19], followed by reshaping operations f_{RF}^B and f_{RF}^{UE} to align the final outputs of the \mathbf{V}_{RF} and \mathbf{W}_{RF} branches with the required matrix forms. Moreover, following [19] we apply

$$f_D^B(\mathbf{V}_D[k]) = \mathbf{V}_D[k] \sqrt{\frac{P}{\text{Tr}(\mathbf{V}_{RF} \mathbf{V}_D[k] \mathbf{V}_D^H[k] \mathbf{V}_{RF}^H)}}, \quad (9)$$

to adjust the signal transmission power, where P is the maximum per-subcarrier transmission power. For training, we follow [4], [18], [19] in adopting a self-supervised approach with the negative of the AIR as the loss function. As shown in [18, Appendix B], minimizing the BCE of the received bits, as in equation (7), is equivalent to maximizing AIR, aligning with the beamforming system's objective. This method can be generalized to multi-user cases by extending the input to include all users' CSI and adopting sum-AIR as the loss function.

B. LoRA Fine-Tuning

In ML-based HBF, the model must adapt to changing channel conditions using minimal computational resources and a small buffer of recent channel samples. LoRA offers an

¹As shown in [19, Section III.A], averaging channel matrix products over all subcarriers is sufficient for optimizing the analog component of the beamforming.

efficient solution by fine-tuning fewer parameters, making it particularly suited for beamforming adjustments in episodically dynamic channels caused by user mobility.

LoRA fine-tunes a low-rank adapter matrix added to each layer of the DNN while keeping the original pre-trained weights fixed [14, Fig. 1]. For convolutional layers, adapter matrices require an additional reshaping operation to align with the layer's filter structure. Note that the upper branch in Fig. 1 contains 2D convolutional layers in the ResNet-A structure, while the lower branch contains 3D convolutional layers in the ResNet-D structure. As proposed in [20], for a 2D convolutional layer with C_{in} input features, C_{out} output channels, and a kernel size of $K_1 \times K_2$, the low-rank adapted weights are represented as $\mathbf{F} = \mathbf{F}_0 + \Delta\mathbf{F}$, where $\mathbf{F}_0 \in \mathbb{R}^{K_1 \times K_2 \times C_{in} \times C_{out}}$ denotes the fixed, pre-trained weight matrix and $\Delta\mathbf{F} = \phi(\mathbf{F}_B \mathbf{F}_A)$ is the low-rank trainable matrix with $\mathbf{F}_B \in \mathbb{R}^{(C_{in} \cdot K_1) \times r}$ and $\mathbf{F}_A \in \mathbb{R}^{r \times (K_2 \cdot C_{out})}$, where r represents the rank. The operator $\phi(\cdot)$ reshapes the product $\mathbf{F}_B \mathbf{F}_A$ to the original layer dimensions, $K_1 \times K_2 \times C_{in} \times C_{out}$. To generalize LoRA to a 3D convolutional layer with kernel size of $K_1 \times K_2 \times K_3$, we absorb the additional kernel dimension by \mathbf{F}_B , resulting in low-rank matrices $\mathbf{F}_B \in \mathbb{R}^{(C_{in} \cdot K_1 \cdot K_2) \times r}$ and $\mathbf{F}_A \in \mathbb{R}^{r \times (K_3 \cdot C_{out})}$. As in the 2D case, the operator $\phi(\cdot)$ reshapes $\mathbf{F}_B \mathbf{F}_A$ to the target dimensions, $K_1 \times K_2 \times K_3 \times C_{in} \times C_{out}$.

To apply LoRA to the DNN structure shown in Fig. 1, we form and add the $\Delta\mathbf{F}$ tensor to the weights of each convolutional layer within the ResNet structure. During offline training, LoRA integrates seamlessly with pre-training, while for online retraining, only low-rank adapters are updated, reducing backpropagation parameters. This ensures rapid adaptation with minimal samples, making LoRA effective for the fine-tuning task. Our results in the next section validate the efficacy of this approach, demonstrating substantial performance gains.

IV. NUMERICAL RESULTS

Inspired by practical scenarios in fifth-generation mobile networks (5G), we assume the BS is equipped with 64 antennas and 8 RF chains transmitting 2 data streams, while the user has 8 antennas and 2 RF chains. Pre-training is performed on 10,000 samples using a CDL channel [16]. Meta-learning is conducted on 20,000 samples across CDL-A (NLOS), CDL-C (NLOS), CDL-D (LOS), and CDL-E (LOS), with 5,000 samples for each channel type. During pre-training of TL-FMF and LoRA and meta-learning of MAML and ANIL, we use the adaptive moment estimation (ADAM) optimization algorithm [21]. During fine-tuning, TL-FMF and LoRA retain ADAM, while MAML and ANIL employ stochastic gradient descent (SGD) to align with the inner-loop structure of meta-learning approaches. In ANIL, adaptation is performed only on the last layer of the neural network—the most influential layer during fine-tuning [12]. The initial learning rate in both cases is set to 10^{-4} . To monitor generalization and prevent overfitting, a validation set of 1,024 samples is used, and early stopping is implemented to improve computational efficiency, limiting training to 20 epochs. For evaluation, we consider two online fine-tuning scenarios in a macro-cell environment.

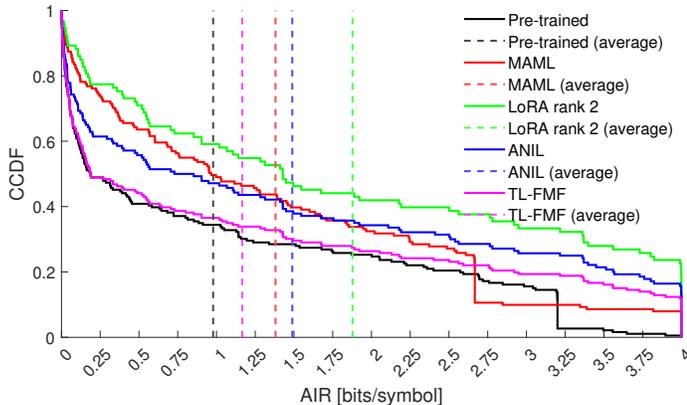


Figure 2. Empirical CCDF (solid lines) and average AIR (dashed lines) of the simulated AIR results for the first online fine-tuning scenario.

First, we simulate an urban area in Munich using NVIDIA’s RT simulator [15], placing a 12-meter base station atop the 98-meter Frauenkirche tower. A user moves randomly within a 400-meter radius of the base station for 1,024 steps. To reduce computational cost, the number of ray-object interactions is limited to 5. For LoRA and TL with FMF (TL-FMF), we use a model pre-trained on a CDL-C channel, while for MAML and ANIL, meta-learning is conducted as previously described. In this simulation, online fine-tuning is performed every 128 steps using the latest 128 channel samples, resulting in a total of 8 fine-tuning updates. In this simulation, the signal and noise spectral power densities are set to -55 dBm/Hz and -174 dBm/Hz, respectively. Second, we conduct a statistical analysis of macro-cell channel transitions across CDL-A, C, D, and E channels, with an average signal-to-noise ratio (SNR) of -5 dB. In these simulations, the signal constellation \mathcal{A} is set to 16-QAM with $K = 1024$ subcarriers. The carrier frequency is 28 GHz, with an antenna spacing of half wavelength and a CDL channel delay spread of 300 ns. The user moves at a speed of 10 m/s in a random direction parallel to the ground, maintaining a constant height of 1.5 meters.

Fig. 2 presents the empirical complementary cumulative distribution function (CCDF) of the simulated AIR results for the first online fine-tuning scenario along with the average AIR achieved by each method. These environment-specific results indicate that LoRA outperforms all other methods, exhibiting a higher frequency of elevated AIR values with a 45% improvement in average AIR over the pre-trained model. The CCDFs of MAML and ANIL intersect around 2 bits/symbol, where MAML demonstrates better performance in avoiding very low AIRs but struggles to achieve higher AIRs compared to ANIL leading to a higher average AIR for ANIL. Also, TL-FMF consistently outperforms the pre-trained model across the range.

In the second scenario, we conduct a more generalizable analysis by extending our simulations to channel transitions in a macro-cell scenario defined by the CDL model (1). Fig. 3 illustrates the AIR performances as functions of the number of trainable parameters for the different CDL model transitions, with results averaged over 4,096 simulations. This

figure includes performances for both a small fine-tuning dataset (128 samples) and a moderate fine-tuning dataset (512 samples) as indicated on the plots. As shown on the x-axis of Fig. 3, the number of trainable parameters—a proxy for the computational cost of online fine-tuning—for the considered methods are 73k for ANIL, 261k/267k/549k for LoRA rank-2/4/8, and 4.268M for MAML and TL-FMF. The results show that LoRA achieves more than 50% improvement over the pre-trained model and consistently outperforms all benchmarks across both dataset sizes. Notably, LoRA with rank-2 matrices performs best for the small 128-sample fine-tuning dataset. An increase in the rank of the adaptation matrices is beneficial only if the fine-tuning dataset size also increases (512 samples vs. 128 samples). This underscores the importance of balancing dataset size with model complexity in LoRA-based fine-tuning. The comparisons show that neither MAML nor ANIL consistently outperforms the other, highlighting differences in their adaptability across scenarios. This observation suggests that ANIL’s previously noted disadvantage in avoiding low AIRs from the RT scenario may not extend across all conditions. However, ANIL’s fewer fine-tunable parameters makes it preferable under computational constraints.

For the second scenario, we also analyzed the catastrophic forgetting (CF) phenomenon, with the results summarized in Table I. The table reports the relative change in AIR when the DNN is trained on CDL-C, subsequently fine-tuned on CDL-A, CDL-D, or CDL-E, and then evaluated on CDL-C, compared to the ideal case where no fine-tuning with a mismatched channel is performed. The results show that MAML experiences the most significant forgetting as its focus on rapid adaptability increases susceptibility to deviations from previous configurations [22]. LoRA and ANIL exhibit stronger resilience to CF by fine-tuning only a small subset of model parameters, preserving prior knowledge while enabling effective adaptation. Notably, rank-2 LoRA outperforms higher-rank configurations, aligning with findings in [23] and further demonstrating its efficiency in mitigating catastrophic forgetting.

Table I
PERFORMANCE DEGRADATION DUE TO FINE-TUNING ON MISMATCHED CHANNELS. PRE-TRAINING IS DONE ON CDL-C.

Method	Finetuned on	CDL-A	CDL-D	CDL-E
ANIL		-9%	-10%	-9%
LoRA-2		-12%	-11%	-14%
LoRA-4		-20%	-23%	-15%
LoRA-8		-17%	-26%	-20%
TL-FMF		-20%	-23%	-21%
MAML		-36%	-27%	-37%

V. CONCLUSION

In this paper, we have proposed LoRA for online adaptation of ML-based HBF in dynamically changing channels. We demonstrated its effectiveness through simulations across a range of channel realizations. In particular, our simulations have shown that LoRA achieves an over 45% improvement in AIR compared to pre-trained models and outperforms

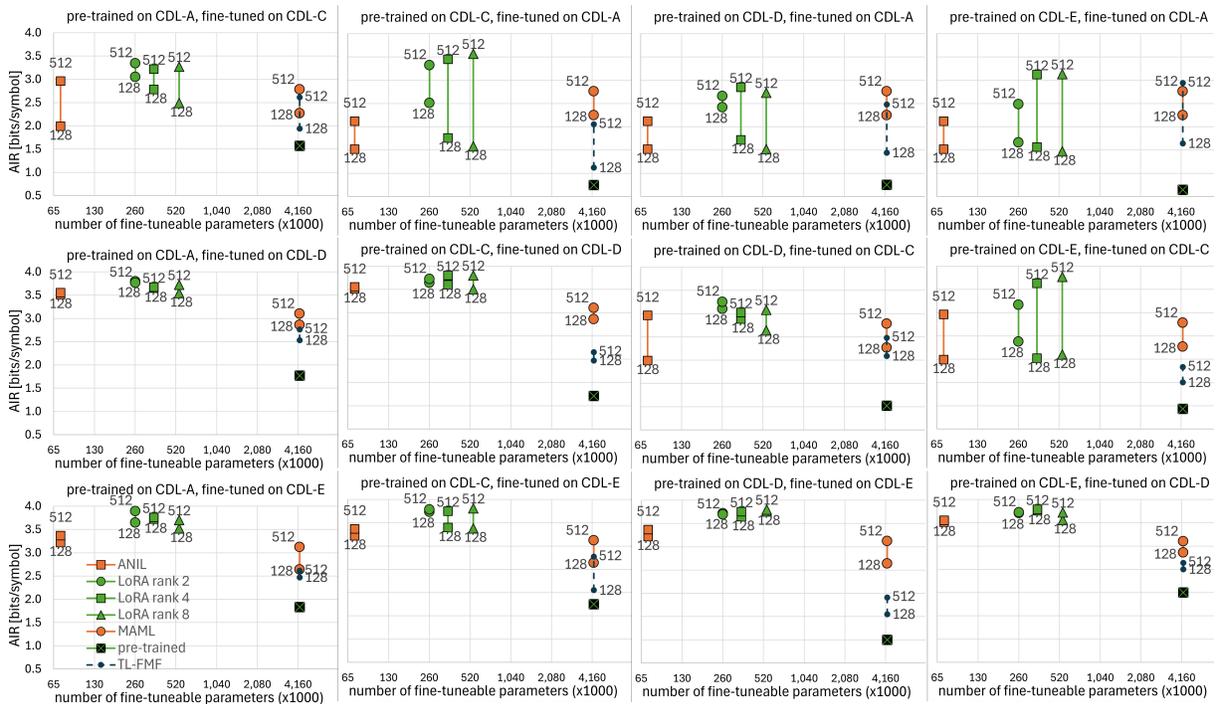


Figure 3. Performance comparison after fine-tuning for channel transitions. Numbers shown on the plots represent the number of fine-tuning samples.

traditional fine-tuning methods, including MAML, ANIL, and TL-FMF. Rank-2 LoRA and ANIL both benefit from a dramatically reduced number of parameters, making them practical choices for online fine-tuning of ML-based HBF optimizers amid channel transitions. However, LoRA consistently achieves a higher AIR in all simulated scenarios while maintaining a low CF, striking a favorable trade-off between complexity and performance.

REFERENCES

- [1] G. M. Zilli and W.-P. Zhu, "Constrained tensor decomposition-based hybrid beamforming for mmWave massive MIMO-OFDM communication systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5775–5788, 2021.
- [2] M. Goutay, F. A. Aoudia, J. Hoydis, and J.-M. Gorce, "Machine learning for MU-MIMO receive processing in OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2318–2332, 2021.
- [3] J. M. J. Huttunen, D. Korpi, and M. Honkala, "DeepTx: Deep learning beamforming with channel prediction," *IEEE Trans. Wireless Commun.*, vol. 22, no. 3, pp. 1855–1867, 2023.
- [4] Q. Hu, Y. Cai, K. Kang, G. Yu, J. Hoydis, and Y. C. Eldar, "Twotimescale end-to-end learning for channel acquisition and hybrid precoding," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 163–181, 2022.
- [5] Z. Gao, M. Wu, C. Hu, F. Gao, G. Wen, D. Zheng, and J. Zhang, "Data-driven deep learning based hybrid beamforming for aerial massive MIMO-OFDM systems with implicit CSI," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 10, pp. 2894–2913, 2022.
- [6] H. Sun, W. Pu, X. Fu, T.-H. Chang, and M. Hong, "Learning to continuously optimize wireless resource in a dynamic environment: A bilevel optimization perspective," *IEEE Trans. Signal Process.*, vol. 70, pp. 1900–1917, 2022.
- [7] Y. Yuan, G. Zheng, K.-K. Wong, B. Ottersten, and Z.-Q. Luo, "Transfer learning and meta learning-based fast downlink beamforming adaptation," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1742–1755, 2021.
- [8] T. Raviv, S. Park, O. Simeone, Y. C. Eldar, and N. Shlezinger, "Online meta-learning for hybrid model-based deep receivers," *IEEE Trans. Wireless Commun.*, vol. 22, no. 10, pp. 6415–6431, 2023.
- [9] H. Zhou, W. Xia, H. Zhao, J. Zhang, Y. Ni, and H. Zhu, "Continual learning-based fast beamforming adaptation in downlink MISO systems," *IEEE Wireless Commun. Lett.*, vol. 12, no. 1, pp. 36–39, 2023.
- [10] A. H. Karkan, H. Hojatian, J.-F. Frigon, and F. Leduc-Primeau, "SAGE-HB: Swift adaptation and generalization in massive MIMO hybrid beamforming," in *IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, 2024, pp. 323–328.
- [11] Q. Hou, M. Lee, G. Yu, and Y. Cai, "Meta-gating framework for fast and continuous resource optimization in dynamic wireless environments," *IEEE Trans. Commun.*, vol. 71, no. 9, pp. 5259–5273, 2023.
- [12] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, "Rapid learning or feature reuse? Towards understanding the effectiveness of MAML," in *International Conference on Learning Representations (ICLR)*, 2020.
- [13] A. Fallah, A. Mokhtari, and A. Ozdaglar, "On the convergence theory of gradient-based model-agnostic meta-learning algorithms," in *Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2020, pp. 1082–1092.
- [14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022.
- [15] NVIDIA, "Sionna: An open-source library for 6G physical-layer research," <https://developer.nvidia.com/sionna>, accessed: 2024-11-5.
- [16] ETSI, "TR 138 901 V18.0.0: Study on channel model for frequencies from 0.5 to 100 GHz," 2024.
- [17] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral energy, and hardware efficiency," *Foundations and Trends in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017.
- [18] F. Ait Aoudia and J. Hoydis, "End-to-end learning for OFDM: From neural receivers to pilotless communication," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 1049–1063, 2022.
- [19] Q. Yuan, H. Liu, M. Xu, Y. Wu, L. Xiao, and T. Jiang, "Deep learning-based hybrid precoding for Terahertz massive MIMO communication with beam squint," *IEEE Commun. Lett.*, vol. 27, pp. 175–179, 2023.
- [20] W. Chen, Z. Miao, and Q. Qiu, "Parameter-efficient tuning of large convolutional models," *arXiv preprint arXiv:2403.00269*, 2024.
- [21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.
- [22] P. Yap, H. Ritter, and D. Barber, "Addressing catastrophic forgetting in few-shot problems," in *International Conference on Machine Learning*, PMLR, 2021, pp. 11909–11919.
- [23] D. Biderman, J. Portes, J. J. G. Ortiz, M. Paul, P. Greengard, C. Jennings, D. King, S. Havens, V. Chiley, J. Frankle *et al.*, "Lora learns less and forgets less," *arXiv preprint arXiv:2405.09673*, 2024.