


Article

Machine Learning Techniques for Blind Beam Alignment in mmWave Massive MIMO

Aymen Ktari *, Hadi Ghauch and Ghaya Rekaya-Ben Othman

Télécom Paris, 91120 Paris, France; hadi.ghauch@telecom-paris.fr (H.G.); ghaya.rekaya@telecom-paris.fr (G.R.-B.O.)

* Correspondence: aymen.ktari@telecom-paris.fr

Abstract: This paper proposes methods for Machine Learning (ML)-based Beam Alignment (BA), using low-complexity ML models, and achieves a small pilot overhead. We assume a single-user massive mmWave MIMO, Uplink, using a fully analog architecture. Assuming large-dimension codebooks of possible beam patterns at *UE* and *BS*, this data-driven and model-based approach aims to partially and blindly sound a small subset of beams from these codebooks. The proposed BA is blind (no CSI), based on Received Signal Energies (RSEs), and circumvents the need for exhaustively sounding all possible beams. A sub-sampled subset of beams is then used to train several ML models such as low-rank Matrix Factorization (MF), non-negative MF (NMF), and shallow Multi-Layer Perceptron (MLP). We provide an extensive mathematical description of these models and the algorithms for each of them. Our extensive numerical results show that, by sounding only 10% of the beams from the *UE* and *BS* codebooks, the proposed ML tools are able to accurately predict the non-sounded beams through multiple transmitted power regimes. This observation holds as the codebook sizes at *UE* and *BS* vary from 128×128 to 1024×1024 .

Keywords: mmWave MIMO; massive antennas; ML-based Beam Alignment; blind BA; Matrix Factorization; Multi-Layer Perceptron; non-linear regression



Citation: Ktari, A.; Ghauch, H.; Rekaya-Ben Othman, G. Machine Learning Techniques for Blind Beam Alignment in mmWave Massive MIMO. *Entropy* **2024**, *26*, 626. <https://doi.org/10.3390/e26080626>

Academic Editors: Eduard Jorswieck, Jaroslaw Krzywanski, Marcin Sosnowski, Karolina Grabowska, Dorian Skrobek and Ghulam Moeen Uddin

Received: 2 April 2024
Revised: 14 June 2024
Accepted: 10 July 2024
Published: 25 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Driven by the explosive growth trend of large-scale connectivity and higher data rate systems, wireless data traffic is expected to exponentially increase, growing to 5 zettabytes per month and reaching a 100 Gps data rate by 2030 [1]. Thus, the latency in the 6th Generation is predicted to reach 0.1 ms, representing 10% of 5G latency, in order to support new emerging technical needs, including holographic images, Internet of Things applications, and autonomous driving.

Beam Alignment is frequently defined in the literature as beam sounding, i.e., beam training. It illustrates a fundamental problem in millimeter-wave Multiple Input, Multiple Output systems, defined as the exchange of information between the user equipment *UE* and the base station *BS* in order to accurately select the optimal beam-steering direction. The process of aligning the beams is related to several technical problems, such as beam forming, beam sweeping, beam tracking, and beam selection. The whole framework that unites these operations between *UE* and *BS* is often denoted as the Beam Management. To fulfill the BA task, beam patterns stored in large codebooks are used at both *UE* and *BS*. In fact, pencil beams with directional gain are increasingly being used in several applications in order to alleviate the severe path-loss attenuation and increase capacity and data throughput. On the other hand, massive MIMO systems provide large gain in spectral and energy efficiencies compared with conventional MIMO systems. Using mmWave technology, these systems mainly offer a better communication quality by increasing the system bandwidth and reducing the effects of noise and interference. Due to the diversification of future 5G and towards 6G applications and intelligent systems, scientists predict the continuous generation of massive datasets for deep processing through large

bandwidths, which introduces mmWave bands as the golden spectrum band candidates. However, the limitations of mmWave communication physical properties of the channel are crucial: scattering, attenuation, low coherence time related to the Doppler effect, penetration loss, environmental constraints, and complex channel modeling in realistic urban scenarios. The major problem we aim to encounter in this paper is the inevitable high signaling/training *overhead*. For this reason, the main trade-off is to browse the most accurate and the least complex ML algorithm that optimizes finding the optimal beam pair based on sounded instantaneous Received Signal Energies and using the minimum (possible) amount of training samples.

Contributions: In this current work, we propose ML-based BA methods, for a single user massive mmWave MIMO, Uplink, with a wide-band channel. We assume a single radio frequency chain with large codebooks of possible analog beams at BS (also known as BS codebook) and UE (also known as UE codebook). We define a beam pair as one beam from the BS and UE codebook. By approximating the SNR with the Receive Signal Energy (RSE), we bypass the need for CSI, i.e., a blind approach. We sub-sample large codebooks into smaller sub-sampled BS and UE codebooks, and sound the beam pairs from the sub-sampled codebooks to generate the training set—a novelty of the approach. Using the RSE of the sounded beam pairs (sub-sampled codebooks), we propose to train the following ML methods to predict the RSE of the beam pairs that were not sounded: Matrix Factorization (MF), non-negative Matrix Factorization (NMF), and feed-forward (shallow) Multi-Layer Perceptron (MLP).

- We formulate the MF and NMF problems. We propose to use Block Coordinate Descent (BCD) and Block Gradient Descent (BGD) methods to solve each problem. We derive in depth all the update equations for these methods. We show that the BCD method converges to a stationary point from both MF and NMF problems. Our extensive numerical results show that, sub-sampling 10% of the BS/UE codebooks, the remaining RSE values can be predicted extremely well (with a training/test error $\approx 10^{-6}$) for every antenna configuration.
- We develop at length the equations of a general MLP model, the resulting loss function, and the corresponding optimization problem. In addition, we derive the equations of back-propagation for the MLP in question. Using extensive numerical results, we observe that sounding 10% of original codebooks is sufficient to predict the RSE of the beam pairs that were not sounded, with negligible training/test error.
- We numerically compare the training/test losses of all the proposed models for a varying cardinality of codebooks and transmit powers. These results suggest that the BCD method for MF/NMF outperforms the MLP in terms of training and test error. Meanwhile, BCD for MF/NMF has a large computational complexity and the MLP exhibits medium complexity.
- Interestingly, by sounding 10% of the BS/UE codebooks, the proposed ML models can predict the unknown RSE (beam pairs not sounded) with a negligible test error. Thus, the proposed methods achieve a 90% reduction in pilot signaling overhead, compared with the SotA benchmark, without any noticeable loss in performance.

Notations: Matrices and vectors are respectively written in boldface upper-case and lower-case letters. We use $\text{Tr}[\mathbf{A}]$, \mathbf{A}^T , \mathbf{A}^{-1} , \mathbf{A}^H , $|\mathbf{A}|$, $\|\mathbf{A}\|_F$ for the trace, transpose, inverse, conjugate transpose, determinant, and Frobenius norm of a matrix \mathbf{A} and the $n \times n$ identity matrix. $[A]_{i,j}$ is used to denote the (i, j)th entry of a matrix \mathbf{A} . We denote the Hadamard product by \circ , while $[\mathbf{a}]_+ := \max(\mathbf{a}, \mathbf{0})$ illustrates a Euclidean projection of \mathbf{a} on \mathbb{R}_+^D and is applied element by element on \mathbf{a} . We denote $|x|$ the absolute value of x and $[x]_t$ as the entry t of a vector \mathbf{x} .

Methods/Experiment: The proposed approach is data driven and model based. The dataset is generated following the Saleh Valenzuela wide-band mmWave system model. It is based on Received Signal Energies for each and every beam pair in the massive MIMO Uplink setup stored in separate .csv files. The model-based solution to the empirical risk minimization includes deriving a closed-form solution to the formulated non-convex

optimization problem, stating the theoretical guarantees of convergence and empirically illustrating the success of the proposed partial and blind Beam Alignment procedure using different algorithms. All simulations are executed on Infres GPU servers and the Comelec laboratory PC at Télécom Paris, having the following characteristics: Intel(R) Core(TM) i5-8365U CPU @ 1.60 GHz, 16 Go (RAM), x64 processor, and 64-bit operating system under the license of Windows 10 Enterprise LTSC 2018, version 1809. The manufacturer is Dell and is located in Paris, France. All python packages used in this work (numpy, scipy, keras, pytorch, matplotlib..) are related to python 3.9 release. In fact, the experimental protocol is based on offline grid-search cross-validation, which requires GPU processing for the selection of optimal hyperparameters and online training/prediction for Matrix Factorization, non-negative Matrix Factorization, and Multi-Layer Perceptron. The comparison is conducted following a Quality of Service-based approach, simulating a variety of MIMO configurations and architectural setups, investigating the impact of varying the Received Signal Energy regime and empirically stating intersections and differences in the impact of the transmit power on model behaviors, loss values, optimal signaling overhead ratio, and optimal hyperparameters.

- **Problem Statement:** The main challenge addressed in this study is the high signaling overhead in Beam Alignment for mmWave MIMO systems, which hampers the efficient selection of optimal beam-steering directions.
- **Research Questions and Hypotheses:** This study investigates whether machine learning methods can effectively reduce the signaling overhead required for accurate beam-pair prediction in mmWave MIMO systems.
- **Objectives and Aims:** The primary objective is to develop and evaluate ML-based BA methods that minimize the training overhead while maintaining high accuracy in predicting the RSE for unsounded beam pairs.
- **Significance and Rationale:** The study proposes a novel approach to BA using ML techniques, which can lead to a substantial reduction in pilot signaling overhead and enhance the efficiency of future wireless communication systems.

2. Literature Survey

In conventional standards, *Exhaustive* BA, also called Brute Force BA, is the de facto approach for the alignment process. It is based on sounding all available beams at both *UE* and *BS* codebooks in order to exhaustively select the optimal beam pair. One obvious drawback is the fact that the resulting signaling overhead scales as the product of the *UE* and *BS* codebook sizes. At 60 GHz, the Exhaustive BA has been adopted in several mmWave *WLAN* or *WPAN* communication technologies, e.g., IEEE 802.15.3c [2] and IEEE 802.11ad [3]. It is conventionally applied in small MIMO configurations using small codebook sizes (e.g., codebooks of size 8×8 for *LTE*) and guarantees optimal performance. For cellular networks [4], V2X communications, Unmanned Aerial Vehicles, or High-Speed Train applications, the infeasibility of brute-force-based BA pushes scientists to reduce the large signaling overhead from using massive antennas systems. State-of-the-art methods can be divided into two categories: classic BA and learning-based BA. Traditional techniques tend to use a more and more structured Beam Alignment design such as hierarchical multi-level codebooks [5] (training beamforming vectors are constructed with different beam widths at different levels) and an overlapped beam pattern [6], where the main idea is to augment the amount of information carried by each channel measurement, reducing the required channel estimation time and beam coding [7], where we assign a unique code signature to each beam angle in addition to subspace estimation/decomposition-based BA [8]. Compressed sensing-based algorithms [9] are also used in this context, taking advantage of channel sparsity. Therefore, we state two intersections in classic methods: they generally rely on *CSI* exchange and Exhaustive BA. In contrast, lately, Machine Learning (*ML*)-based BA has emerged and is continuously leading to some promising results. For instance, statistical models such as Kolmogorov model-based BA in [10] with sub-sampled codebooks reduce the signaling overhead: 15% of Exhaustive BA provides accurate predictions for

optimal beams at *UE* and *BS* in a partial *BA* procedure, similar to our approach. Deep learning through shallow neural networks is increasingly used by Wireless Communication scientists, where we distinguish two major paradigms: first, the ML methods related to Supervised Learning (*SL*) via a Support Vector Machine and Multi-Layer Perceptrons for joint analog beam selection in [11], convolutional neural networks for beam management in sub-6 GHz in [12] and for calibrated beam training in [13], recurrent neural networks such as Long Short-Term Memory network for beam tracking in [14–16], auto-encoders for beam management in [17], and several other neural architectures, and second, Reinforcement Learning (*RL*) in [18–20], generally used to resolve the problems of Multi-Armed Bandit and Markov decision process. In addition, neural architectures have the ability to extract features from the hidden interactions between *BS* and *UE*, providing fast and accurate estimations through different MIMO setups and channel realizations, especially when applied to massive datasets where more and more data/train samples are embedded. This work is an extension of [21]. In this paper, we extend the channel model to wide-band and we add multiple RF-chains at *BS* in a fully analog low-complexity architecture, where we investigate more *ML* tools for partial and blind *BA*. This paper is one of the first attempts to apply *MF/NMF* models and shallow Multi-Layer Perceptrons to a blind and partial Beam Alignment for massive mmWave SU-MIMO. Our work in [22] is related to the same approach and objectives, where we quantize the output of each RF-chain.

3. System Model

In this section, we illustrate the mmWave MIMO point-to-point system model. We consider an Uplink transmission from multiple-antenna user equipment *UE* using a single radio frequency chain and a multiple-antenna base station *BS* using multiple radio frequency chains. The proposed *ML* methods are performed at the *BS*, which has higher computational resources than *UE*. Figure 1a,b provide a diagram representation of the proposed architecture. *UE* and *BS* are respectively equipped with Uniform Linear Arrays of N_T and N_R antenna. We propose a low-cost/complexity fully analog architecture where *UE* has one radio frequency chain and *BS* has N_{rf} radio frequency chains. *UE* selects its analog beamformer $\mathbf{f}_u \in \mathbb{C}^{N_T}$ from a codebook of feasible beam choices, $u \in \mathcal{T}$, where \mathcal{T} is the corresponding index set. Moreover, *BS* selects its analog combiner $\mathbf{w}_i \in \mathbb{C}^{N_R \times N_{rf}}$ from a codebook $i \in \mathcal{R}$ with \mathcal{R} as the index set of the codebook. We denote with C_T the number of possible beamforming vectors at *UE*, i.e., the size/cardinality of the *UE* codebook, $|\mathcal{T}| = C_T$ and C_R , and the size/cardinality of the *BS* codebook, $|\mathcal{R}| = C_R$. Both beamforming and combining are fully performed in the analog domain using phase shifters at *UE* and *BS*; thus, they satisfy the following constant modulus constraints, $\forall r \in \{1, \dots, N_R\}, \forall t \in \{1, \dots, N_{rf}\}$:

$$\mathbf{w}_i \in \mathbb{C}^{N_R \times N_{rf}}, \quad |[w_i]_{r,t}| = \frac{1}{N_{rf}N_R}$$

$$\mathbf{f}_u \in \mathbb{C}^{N_T}, \quad |[f_u]_t| = \frac{1}{N_T}, \quad \forall t \in \{1, \dots, N_T\}$$

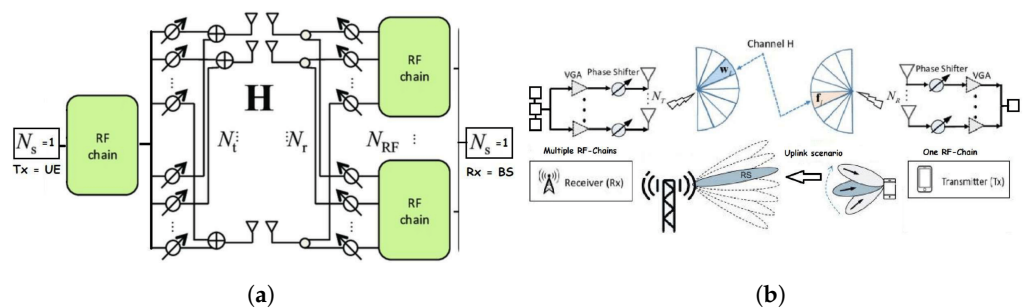


Figure 1. Proposed *BA* diagram representation: (a) fully analog MIMO architecture using a single RF chain at *UE* and multiple RF chains at *BS*; (b) simplified illustration of Beam Alignment problem.

For our proposed approach, BS is responsible for receiving signal energies, denoted as RSE , in order to learn their patterns and features for the purpose of accurately predicting the optimal beam indexes from their corresponding codebooks and send them to UE. We adopt the wide-band channel model $\mathbf{G} \in \mathbb{C}^{N_R \times N_T}$ given by

$$\mathbf{G}(k) = \sqrt{\frac{1}{N_c}} \sum_{l=1}^{N_c} \mathbf{H}_l e^{-j2\pi lk/N_c}, \forall k \in \{1, \dots, N_c\} \tag{1}$$

where N_c represents the number of sub-carriers over the whole bandwidth through an OFDM scenario, k is the index of the sub-carrier k , and $\mathbf{H}_l \in \mathbb{C}^{N_R \times N_T}$ is the narrow band channel model representing the time domain channel impulse response with L -tapped delays given by $\mathbf{H}_l = \sqrt{\frac{N_T N_R}{L}} \sum_{i=1}^L \rho_i \mathbf{a}_R(\theta_i^{(R)}) \mathbf{a}_T^H(\theta_i^{(T)})$, where L is number of paths (rank) of the channel; $\theta_i^{(R)}$ and $\theta_i^{(T)}$ are the angles of arrival at BS and the angles of departure from UE, noting AoA/AoD to correspond to the i^{th} path (and both assumed to be uniform over $[-\pi/2, \pi/2]$); ρ_i is the complex gain of the i^{th} path such that $\rho_i \sim \mathcal{CN}(0, 1)$, $\forall i$; and last but not least, $\mathbf{a}_R(\theta_i^{(R)}) \in \mathbb{C}^{N_R}$ and $\mathbf{a}_T(\theta_i^{(T)}) \in \mathbb{C}^{N_T}$ are the array response vectors at both UE and BS, respectively. We further assume that the channel is completely unknown to both UE and BS. Henceforth, in this paper, we shall denote the beam pair (u, i) as the combination of the UE beamformer indexed u from the UE codebook \mathcal{T} and combiner indexed i in the BS codebook \mathcal{R} . The signal at BS resulting from applying the beam pair (u, i) , $\mathbf{y}_{u,i} \in \mathbb{C}^{N_{rf}}$ is expressed as

$$\mathbf{y}_{u,i} = \mathbf{W}_i^H \mathbf{G}(k) \mathbf{f}_u s_u + \mathbf{n}_i, \forall (u, i) \in \mathcal{T} \times \mathcal{R}, \tag{2}$$

where $s_u = 1/\sqrt{P_u}$ is the transmitted pilot symbol associated with \mathbf{f}_u (having power $\sqrt{P_u}$) and $\mathbf{n}_i = \mathbf{W}_i^H \mathbf{n}$ is the effective additive white Gaussian noise AWGN with unit variance ($\sigma^2 = 1$). We define the received Signal-to-Noise Ratio (SNR) for the beam pair (u, i) as $SNR_{u,i} = P_u \|\mathbf{W}_i^H \mathbf{G}(k) \mathbf{f}_u\|_2^2, \forall (u, i) \in \mathcal{T} \times \mathcal{R}$. We assume a fully blind approach; i.e., neither BS nor UE has any knowledge of \mathbf{G} . Thus, computing the above SNR expression is not feasible due to the fact that BS is assumed not to know \mathbf{G} . Thus, in this work, we will approximate the SNR of the beam pair (u, i) using the corresponding instantaneous Received Signal Energies (RSEs) expressed as $RSE_{u,i} = \|\mathbf{y}_{u,i}\|_2^2, \forall (u, i) \in \mathcal{T} \times \mathcal{R}$. In other words, we will assume that $RSE_{u,i} \approx SNR_{u,i}$ for each beam pair $(u, i) \in \mathcal{T} \times \mathcal{R}$.

Benchmark: Exhaustive BA: The de facto method for Beam Alignment is Exhaustive BA. It is accomplished by *exhaustively sounding*, jointly, the beams of both UE and BS codebooks, recording all entries of RSE, and exhaustively searching \mathbf{S} for the indexes of the beam pair that maximize RSE at BS, i.e., $(u^*, i^*) = \underset{(u,i) \in \mathcal{T} \times \mathcal{R}}{\operatorname{argmax}} RSE_{u,i}$. Thus, the RSE matrix is com-

puted/recorded N_{rf} -entries, with each of pilot symbol, since N_{rf} samples are simultaneously received at the BS for every pilot transmission (see Figure 2). Consequently, the pilot signaling overhead of the Exhaustive BA is $\Omega = |\mathcal{T} \times \mathcal{R}| / N_{rf} = C_T C_R / N_{rf}$, which implies that the overhead of this benchmark scales poorly with the BS and UE codebooks.

Proposed partial Beam Alignment using sub-sampled codebooks: Recall the designation of the *beam pair* (u, i) as the beamforming vector of the index u in the UE codebook of beams and the combining vector of the index i in the BS codebook of beams. First, we select (at random) the indexes of the *sub-sampled codebooks* of beams at UE and BS, \mathcal{R}_S and \mathcal{T}_S , such that $\mathcal{R}_S \subset \mathcal{R}$ and $\mathcal{T}_S \subset \mathcal{T}$, and $|\mathcal{R}_S| \ll |\mathcal{R}|$ $|\mathcal{T}_S| \ll |\mathcal{T}|$. The idea behind this approach is to only sound beam pairs from the sub-sampled codebook of beams, \mathcal{R}_S and \mathcal{T}_S . We thus define the *training set*, \mathcal{K} , as the sub-sampled codebook indexes at UE and BS, i.e., $\mathcal{K} := \{(u, i) \mid (u, i) \in \mathcal{T}_S \times \mathcal{R}_S\}$. Then, the RSE of the sounded beam pairs (training set) is given to several ML methods, and the learned ML model is used to predict the RSE of non-sounded beam pairs.

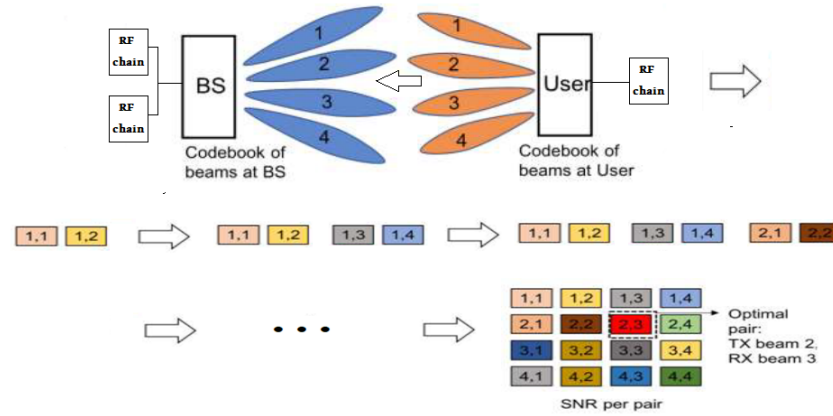


Figure 2. Exhaustive Beam Alignment: $|\mathcal{T}| = |\mathcal{R}| = 4$, $N_{rf} = 2$ RF-Chains at BS. Record 2 beam pairs for each pilot symbol transmission until the matrix is complete. Signaling overhead, $\Omega = \frac{4 \times 4}{2}$.

We formalize this proposed method below. We express both the received signal $\mathbf{y}_{(u,i)}$ and RSE for the beam pair (u, i) resulting from the sounded beam pairs (i.e., training set), as follows:

$$\mathbf{y}_{u,i} = \mathbf{W}_i^H \mathbf{G}(k) \mathbf{f}_{uS} + \mathbf{n}_i, \forall (u, i) \in \mathcal{T}_S \times \mathcal{R}_S \quad (3)$$

$$RSE_{u,i} = \|\mathbf{y}_{u,i}\|_2^2, \forall (u, i) \in \mathcal{T}_S \times \mathcal{R}_S. \quad (4)$$

The dataset is formulated using the following incomplete RSE matrix, $\mathbf{S} \in \mathbb{R}^{C_T \times C_R} (:= \mathbb{R}^{|\mathcal{T}| \times |\mathcal{R}|})$:

$$[\mathbf{S}]_{u,i} := \begin{cases} RSE_{u,i} & , \text{ if } (u, i) \in \mathcal{T}_S \times \mathcal{R}_S \\ \text{Unknown RSE} & , \text{ if } (u, i) \notin \mathcal{T}_S \times \mathcal{R}_S \end{cases} \quad (5)$$

where $[\mathbf{S}]_{u,i}$ denotes the element (u, i) of \mathbf{S} , $\forall (u, i) \in \mathcal{T} \times \mathcal{R}$. Evidently, the value of RSE is undefined for the beam pairs that were not sounded, designated as unknown-RSE matrix coefficient. Those are the missing entries, which are predicted using one of the following proposed ML methods: (i) low-rank MF/NMF and (ii) shallow (feed-forward) MLP, where we utilize the sounded RSE entries as the training set, \mathcal{K} . Then the training set, \mathcal{K} , is fed into one of the above ML models, which will predict the RSE of non-sounded coefficients in \mathbf{S} , denoted as ‘Unknown’, in (5) (see Figure 3). Finally, the pilot signaling overhead for the above-proposed sub-sampled codebook method is $\Omega = |\mathcal{T}_S \times \mathcal{R}_S| / N_{rf} = |\mathcal{K}| / N_{rf}$. We split the RSE dataset into a training set \mathcal{K} and a test set \mathcal{L} such that $\mathcal{K} \cap \mathcal{L} = \{\}$. In this paper, $RSE_{u,i}$ denotes the true value (label) of the RSE for the beam pair (u, i) in the training set \mathcal{K} , and $\widehat{RSE}_{u,i}$ denotes the true value (label) of the RSE for the beam pair (u, i) in the test set \mathcal{L} .

Signaling overhead ratio: It is defined as $\eta := \frac{\text{overhead of learning-based BA}}{\text{overhead of Exhaustive BA}} = \frac{|\mathcal{T}_S| \times |\mathcal{R}_S|}{|\mathcal{T}| \times |\mathcal{R}|} = \frac{|\mathcal{K}|}{C_T C_R}$, where \mathcal{T}_S and \mathcal{R}_S are, respectively, the sizes of the UE and BS sub-sampled codebooks used in our proposed partial beam sounding, while \mathcal{T} and \mathcal{R} refer to the original size of the codebooks, and $0 < \eta \leq 1$ measures the signaling overhead of all the proposed MF, MLP, and AE methods compared with that of Exhaustive BA. Evidently, a small value for η is desired to reduce the signaling overhead of our proposed method. However, a low η implies that the size of the training set is small. As a result, the proposed ML method will not be able to extract enough data patterns due to the (too) small number of training samples, resulting in a larger prediction error. As one of the contributions of this work, we will (empirically) find as small a value for η as possible while still having extremely small training and prediction error.

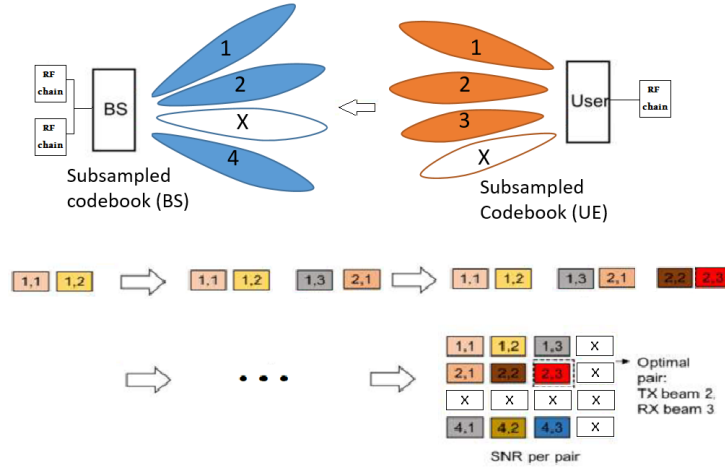


Figure 3. Proposed partial Beam Alignment using sub-sampled codebooks: $|\mathcal{T}| = |\mathcal{R}| = 4$, $N_{rf} = 2$ RF-Chains: record 2 beam pairs for each pilot symbol transmission until sounded beams are recorded. The missing entries represent the predicted entries. Signaling overhead, $\Omega = \frac{3 \times 3}{2}$.

Conjecture: Note that, from the equations of the narrow-band channel model \mathbf{H} and the wide-band channel model $\mathbf{G}(k)$, it is simple to verify that $\text{rank}(\mathbf{H}) \leq L$ and $\text{rank}(\mathbf{G}(k)) \leq LN_C$. Assume that $P_u \rightarrow \infty$. Thus, we can approximate the RSE matrix as

$$[\mathbf{S}]_{u,i} = \|\mathbf{y}_{u,i}\|_2^2 = \|\mathbf{W}_i^H \mathbf{G}(k) \mathbf{f}_u \sqrt{P_u} + \mathbf{n}_i\|_2^2 \stackrel{P_u \rightarrow \infty}{\approx} P_u \|\mathbf{W}_i^H \mathbf{G}(k) \mathbf{f}_u\|_2^2, \forall (u,i) \in \mathcal{T} \times \mathcal{R} \quad (6)$$

If $P_u \rightarrow \infty$, then it can be shown that the RSE matrix \mathbf{S} is such that $\text{rank}(\mathbf{S}) \leq LN_C$. This implies that if $P_u \rightarrow \infty$, then $\mathbf{S} \in \mathbb{R}^{C_R \times C_T}$ is a low-rank matrix, i.e., $\text{rank}(\mathbf{S}) \leq LN_C \ll \min(C_T, C_R)$.

While the proof for this necessary condition eludes the authors, we empirically observed that if P_u is large, then the number of non-zero singular values of \mathbf{S} , $\{\sigma_i(\mathbf{S})\}_{i=1}^{\text{rank}(\mathbf{S})}$, satisfies the above upper bound, i.e., $|\{\sigma_i(\mathbf{S})\}_{i=1}^{\text{rank}(\mathbf{S})}| \leq LN_C$.

Remark 1. Recall the expression for the effective rate, r , $r = (1 - \frac{\Omega}{T}) \log(1 + RSE_{u,i})$, where Ω is the pilot signaling overhead and T is the number of symbols per block. Thus, the problem of maximizing r is written as the following series of equivalent problems:

$(u^*, i^*) := \arg \max_{(u,i) \in \mathcal{T} \times \mathcal{R}} r \Leftrightarrow \arg \max_{(u,i) \in \mathcal{T} \times \mathcal{R}} \log(1 + RSE_{u,i}) \Leftrightarrow \arg \max_{(u,i) \in \mathcal{T} \times \mathcal{R}} RSE_{u,i}$, where the last \Leftrightarrow is due to the fact that the $\log(x)$ is a strictly monotonically increasing function in x . This result implies finding the optimal beam pair (u^*, i^*) that maximizes r is equivalent to finding the best beam pair that maximizes the RSE.

Remark 2. The information (number of entries) needed to represent the RSE matrix $\mathbf{S} \in \mathbb{C}^{C_R \times C_T}$ is measured as $\text{rank}(\mathbf{S})(1 + C_T + C_R)$. This result is evident from performing the SVD on \mathbf{S} and counting the resulting number of entries. Thus, if \mathbf{S} is severely rank deficient, i.e., extremely compressible, then methods such as MF/NMF will exhibit extremely small training and test error. Conversely, if \mathbf{S} is full rank, i.e., not compressible, then the training and test of MF/NMF will be quite large.

4. Matrix Factorization and Non-Negative Matrix Factorization

4.1. MF and NMF Problem Formulation

The intuition behind low-rank MF is to model the RSE of the sounded beam pairs (i.e., entries of \mathbf{S} that are known as $\mathcal{T}_S \times \mathcal{R}_S$) as an inner product between two D -dimensional latent vectors/factors, $\boldsymbol{\theta}_u, \boldsymbol{\psi}_i$, as illustrated in Figure 4. Specifically, the RSE of the beam pair (u, i) , denoted as $[\mathbf{S}]_{u,i}$, is modeled as $[\mathbf{S}]_{u,i} := \boldsymbol{\theta}_u^T \boldsymbol{\psi}_i$, $\boldsymbol{\theta}_u \in \mathbb{R}^D$, $\boldsymbol{\psi}_i \in \mathbb{R}^D$, $\forall (u, i) \in \mathcal{K} := \mathcal{T}_S \times \mathcal{R}_S$, where D is the size/dimension/complexity of the Matrix Factorization model latent factors and $\boldsymbol{\theta}_u \in \mathbb{R}^D$, $\boldsymbol{\psi}_i \in \mathbb{R}^D$ are the MF model parameters (to be optimized). In addition, due to the low-rank MF model, D is assumed to be much smaller than the

dimensions of \mathbf{S} , i.e., $D \ll (C_T, C_R)$. The RSE of the beam pair (u, i) is known from sounding the sub-sampled codebooks (i.e., label). The general formulation of our loss function $\ell_{u,i}$ describes the distance between the true value $RSE_{u,i}$ and the predicted value $\theta_u^T \psi_i$, which corresponds to the MF output/prediction: $\ell_{u,i} := (RSE_{u,i} - \theta_u^T \psi_i)^2, \forall (u, i) \in \mathcal{K} (:= \mathcal{T}_S \times \mathcal{R}_S)$. The Empirical Risk (also known as training error) is defined as the average across all the individual loss function $\ell_{u,i}$. We define the regularized Empirical Risk function as the above empirical risk in addition to the following regularization terms:

$$\sum_{(u,i) \in \mathcal{K}} \left[\frac{1}{|\mathcal{K}|} \left([S]_{u,i} - \theta_u^T \psi_i \right)^2 + \lambda_i \|\psi_i\|_2^2 + \mu_u \|\theta_u\|_2^2 \right] = f((\theta_u, \psi_i)_{(u,i) \in \mathcal{K}}) \tag{7}$$

where $\{\lambda_i \geq 0, \mu_u \geq 0 \mid \forall (u, i) \in \mathcal{K}\}$ is the set of regularization hyperparameters used to balance the MF/NMF model, preventing any overfitting or underfitting. The Empirical Risk Minimization corresponding to the MF model is given by

$$(P1) := \{\hat{\theta}_u, \hat{\psi}_i\} \begin{cases} \operatorname{argmin} & f(\theta_u, \psi_i) \\ \{\theta_u, \psi_i\}_{(u,i) \in \mathcal{K}} \\ \text{s. t. } & \theta_u \in \mathbb{R}^D, \psi_i \in \mathbb{R}^D \end{cases}$$

For the Matrix Factorization variant NMF , the optimization problem is given by

$$(P2) := \{\hat{\theta}_u, \hat{\psi}_i\} \begin{cases} \operatorname{argmin} & f(\theta_u, \psi_i) \\ \{\theta_u, \psi_i\}_{(u,i) \in \mathcal{K}} \\ \text{s. t. } & \theta_u \in \mathbb{R}_+^D, \psi_i \in \mathbb{R}_+^D \end{cases}$$

where $\{\hat{\theta}_u, \hat{\psi}_i\}$ denotes the optimal latent vectors for MF and NMF . The test loss (also known as test error) is given by applying the general loss on the unknown data samples (non-sounded beams) using optimal MF/NMF parameters $\hat{\theta}_u$ and $\hat{\psi}_i$: $= \frac{1}{|\mathcal{L}|} \sum_{(u,i) \in \mathcal{L}} \left(\widehat{RSE}_{u,i} - \hat{\theta}_u^T \hat{\psi}_i \right)^2$, where \mathcal{L} is the test set of our learning model.

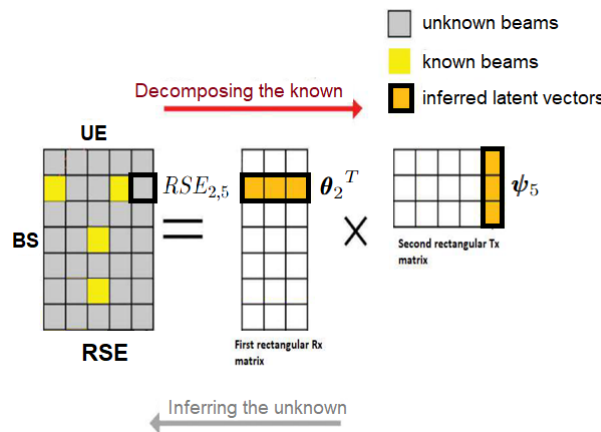


Figure 4. Toy Example: Matrix Factorization with $|\mathcal{T}| = 5, |\mathcal{R}| = 7, D = 3$. MF results into two rectangular matrices to be optimized: MF uses the RSE of known beams (yellow) to predict/complete unknown beams (gray). The product of the latent factors θ_2^T and ψ_5 gives the unknown value of $RSE_{2,5}$.

4.2. Solutions for MF

We resolve the MF problem $(P1)$ using the following methods: (i) Block Coordinate Descent (BCD) often denoted as Alternating Least Squares (ALSs), (ii) BCD with Stochastic Gradient Descent, and (iii) Block Gradient Descent (BGD), which merges BCD and Gradient Descent (GD) definitions.

BCD for MF (BCD MF): BCD proceeds by splitting the optimizing problem $(P1)$ into sub-problems, supposing that all other blocks are known/fixed. We will show that

each sub-problem is strongly convex in each block, and the BCD algorithm converges to a stationary point. The application of BCD to the MF problem results in two sub-problems, S1 and S2, which are solved iteratively. At iteration k , the sub-problem (S1) is defined by fixing the block $\{\boldsymbol{\psi}_i^{(k)}\}_{\forall i}$ and the update/solve block $\{\boldsymbol{\theta}_u\}_{\forall u}$ only, as follows:

$$\begin{aligned} (S1) : \boldsymbol{\theta}_u^{(k+1)} &= \operatorname{argmin}_{\boldsymbol{\theta}_u \in \mathbb{R}^d} f(\{\boldsymbol{\theta}_u, \boldsymbol{\psi}_i^{(k)}\}) \\ &= \sum_{(u,i) \in \mathcal{K}} [([\mathbf{S}]_{u,i} - \boldsymbol{\theta}_u^T \boldsymbol{\psi}_i^{(k)})^2 + \mu_u \|\boldsymbol{\theta}_u\|_2^2 + \lambda_i \|\boldsymbol{\psi}_i^{(k)}\|_2^2] \end{aligned}$$

Moreover, the sub-problem (S2) is defined by fixing the block $\{\boldsymbol{\theta}_u^{(k+1)}\}_{\forall u}$ in (P_1) and the update/solve block $\{\boldsymbol{\psi}_i\}_{\forall i}$, only, as follows:

$$\begin{aligned} (S2) : \boldsymbol{\psi}_i^{(k+1)} &= \operatorname{argmin}_{\boldsymbol{\psi}_i \in \mathbb{R}^d} f(\{\boldsymbol{\theta}_u^{(k+1)}, \boldsymbol{\psi}_i\}) \\ &= \sum_{(u,i) \in \mathcal{K}} [([\mathbf{S}]_{u,i} - \boldsymbol{\theta}_u^{(k+1)T} \boldsymbol{\psi}_i)^2 + \mu_u \|\boldsymbol{\theta}_u^{(k+1)}\|_2^2 + \lambda_i \|\boldsymbol{\psi}_i\|_2^2] \end{aligned}$$

We will rewrite S1 into as series of equivalent problems as follows:

$$\begin{aligned} (S1) &:= \operatorname{argmin}_{\boldsymbol{\theta}_u \in \mathbb{R}^d} \sum_{(u,i) \in \mathcal{K}} [[\mathbf{S}]_{u,i}^2 - 2[\mathbf{S}]_{u,i} \boldsymbol{\theta}_u^T \boldsymbol{\psi}_i^{(k)} + \boldsymbol{\theta}_u^T \boldsymbol{\psi}_i^{(k)} \boldsymbol{\theta}_i^{(k)T} \boldsymbol{\theta}_u + \mu_u \|\boldsymbol{\theta}_u\|_2^2] \\ &\Leftrightarrow \operatorname{argmin}_{\boldsymbol{\theta}_u \in \mathbb{R}^d} \sum_u [-2\boldsymbol{\theta}_u^T \sum_i ([\mathbf{S}]_{u,i} \boldsymbol{\psi}_i^{(k)}) + \boldsymbol{\theta}_u^T \sum_i (\boldsymbol{\psi}_i^{(k)} \boldsymbol{\psi}_i^{(k)T}) \boldsymbol{\theta}_u + \mu_u \|\boldsymbol{\theta}_u\|_2^2] \\ &\Leftrightarrow \operatorname{argmin}_{\boldsymbol{\theta}_u \in \mathbb{R}^d} \sum_{u \in \mathcal{U}_i} [-2\boldsymbol{\theta}_u^T (\mathbf{r}_u^{(k)}) + \boldsymbol{\theta}_u^T (\mathbf{Q}_u^{(k)}) \boldsymbol{\theta}_u + \mu_u \|\boldsymbol{\theta}_u\|_2^2] = \sum_{u \in \mathcal{U}_i} h_u(\boldsymbol{\theta}_u), \\ \boldsymbol{\theta}_u^{(k+1)} &= \operatorname{argmin}_{\boldsymbol{\theta}_u \in \mathbb{R}^d} [-2\boldsymbol{\theta}_u^T \mathbf{r}_u^{(k)} + \boldsymbol{\theta}_u^T (\mathbf{Q}_u^{(k)} + \mu_u \mathbf{I}_D) \boldsymbol{\theta}_u] = f_1(\boldsymbol{\theta}_u), \quad \forall u \in \mathcal{U}_i, \end{aligned}$$

where \mathcal{U}_i is the set of row indexes u in the RSE matrix corresponding to the column i in the known entries of the RSE matrix, $\mathbf{Q}_u^{(k)} = \sum_i (\boldsymbol{\psi}_i^{(k)} \boldsymbol{\psi}_i^{(k)T})$ and $\mathbf{r}_u^{(k)} = \sum_i ([\mathbf{S}]_{u,i} \boldsymbol{\psi}_i^{(k)})$. We derive the closed-form solution for the sub-problem S1 by finding the global min of $f_1(\boldsymbol{\theta}_u)$, as follows:

$$\nabla f_1(\boldsymbol{\theta}_u) = \mathbf{0} \Leftrightarrow -2\mathbf{r}_u^{(k)} + 2(\mathbf{Q}_u^{(k)} + \mu_u \mathbf{I}_D) \boldsymbol{\theta}_u = \mathbf{0} \Leftrightarrow \boldsymbol{\theta}_u = (\mathbf{Q}_u^{(k)} + \mu_u \mathbf{I}_D)^{-1} \mathbf{r}_u^{(k)}$$

Similarly, we rewrite the sub-problem (S2) into the following series of equivalent problems by stating the last one:

$$(S2) : \boldsymbol{\psi}_i^{(k+1)} = \operatorname{argmin}_{\boldsymbol{\psi}_i \in \mathbb{R}^d} [-2\mathbf{t}_i^{(k+1)T} \boldsymbol{\psi}_i + \boldsymbol{\psi}_i^T (\mathbf{P}_i^{(k+1)} + \lambda_i \mathbf{I}) \boldsymbol{\psi}_i] = f_2(\boldsymbol{\psi}_i), \quad \forall i \in \mathcal{I}_u,$$

where \mathcal{I}_u is the set of column indexes i in the RSE matrix corresponding to the row u in the known entries of the RSE matrix, $\mathbf{t}_i^{(k+1)} = \sum_u ([\mathbf{S}]_{u,i} \boldsymbol{\theta}_u^{(k+1)T})$ and $\mathbf{P}_i^{(k+1)} = \sum_u (\boldsymbol{\theta}_u^{(k+1)} \boldsymbol{\theta}_u^{(k+1)T})$. Next, we derive a closed-form solution for the sub-problem S2 by finding the global min of $f_2(\boldsymbol{\psi}_i)$, as follows:

$$\begin{aligned} \nabla f_2(\boldsymbol{\psi}_i) = \mathbf{0} &\Leftrightarrow -2\mathbf{t}_i^{(k+1)} + 2(\mathbf{P}_i^{(k+1)} + \lambda_i \mathbf{I}_D) \boldsymbol{\psi}_i = \mathbf{0} \Leftrightarrow \boldsymbol{\psi}_i = (\mathbf{P}_i^{(k+1)} + \lambda_i \mathbf{I}_D)^{-1} \mathbf{t}_i^{(k+1)} \\ &\Leftrightarrow \boldsymbol{\psi}_i^{(k+1)} = ((\sum_u (\boldsymbol{\theta}_u^{(k+1)} \boldsymbol{\theta}_u^{(k+1)T})) + \lambda_i \mathbf{I}_D)^{-1} (\sum_u ([\mathbf{S}]_{u,i} \boldsymbol{\theta}_u^{(k+1)T})) \end{aligned}$$

Thus, BCD updates to solve MF are given as follows:

$$\begin{cases} \boldsymbol{\theta}_u^{(k+1)} &= (\sum_i \boldsymbol{\psi}_i^{(k)} (\boldsymbol{\psi}_i^{(k)})^T + \mu_u \mathbf{I})^{-1} (\sum_i [\mathbf{S}]_{u,i} \boldsymbol{\psi}_i^{(k)}) \\ \boldsymbol{\psi}_i^{(k+1)} &= ((\sum_u \boldsymbol{\theta}_u^{(k+1)} (\boldsymbol{\theta}_u^{(k+1)})^T) + \lambda_i \mathbf{I})^{-1} (\sum_u [\mathbf{S}]_{u,i} \boldsymbol{\theta}_u^{(k+1)}) \end{cases}$$

$$\forall (u, i) \in \mathcal{K}, k = 0, 1, \dots, I_M \tag{8}$$

where (k) represents the index of the BCD iterations, (u, i) are the codebook indexes at UE and BS, and $[\mathbf{S}]_{u,i}$ denotes the RSE of the (u, i) beam couple. The solution $\{\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\psi}}_i\}_{(u,i) \in \mathcal{K}}$ is reached after the interval/gap between consecutive iterations reaches a predefined ϵ or a max number of iterations, I_M . We have the following result.

Corollary 1. *The sequence of updates $\{\boldsymbol{\theta}_u^{(k)}, \boldsymbol{\psi}_i^{(k)} \mid \forall (u, i) \in \mathcal{K}\}_k$ generated by BCD, in (8), is non-increasing (in k) and converges to a stationary point as $k \rightarrow \infty$.*

Proof. See Appendix A. \square

Block Stochastic Gradient Descent (BSGD) for MF (SGD MF): SGD MF proceeds by taking T plain SGD steps (mini-batch size = 1). BGD proceeds by taking T SGD steps for each block BCD. We first choose at random a single training sample $(u, i) \in \mathcal{K}$. The BSGD update for the sub-problem (S1) is done by performing SGD for $f_1(\boldsymbol{\theta}_u) = \sum_{u \in \mathcal{U}_i} h_u(\boldsymbol{\theta}_u)$, i.e., choosing at random a single index $u \in \mathcal{U}_i$ and computing the plain SGD $\widehat{\nabla f_1(\boldsymbol{\theta}_u)} = \widehat{\nabla}(\sum_{u \in \mathcal{U}_i} h_u(\boldsymbol{\theta}_u)) = h_u(\boldsymbol{\theta}_u)$, where u is a random index from \mathcal{U}_i , and $\widehat{\nabla f_1(\boldsymbol{\theta}_u)}$ is the plain SGD on $f_1(\cdot)$. The corresponding update is given as

$$\begin{aligned} \boldsymbol{\theta}_u^{(k+1)} &= \boldsymbol{\theta}_u^{(k)} - \alpha_k \widehat{\nabla f_1(\boldsymbol{\theta}_u^{(k)})}, = \boldsymbol{\theta}_u^{(k)} - \alpha_k \nabla h_u(\boldsymbol{\theta}_u^{(k)}) \quad u \in \mathcal{U}_i \\ &= \boldsymbol{\theta}_u^{(k)} + 2\alpha_k ((\sum_i([\mathbf{S}]_{u,i} \boldsymbol{\psi}_i^{(k)})) - ((\sum_i \boldsymbol{\psi}_i^{(k)} \boldsymbol{\psi}_i^{(k)T}) + \mu_u \mathbf{I}_D) \boldsymbol{\theta}_u^{(k)}), \quad u \in \mathcal{U}_i, k = 1..T \end{aligned}$$

where u is a single index chosen at random from \mathcal{U}_i , $\mathbf{Q}_u^{(k)} = \sum_i(\boldsymbol{\psi}_i^{(k)} \boldsymbol{\psi}_i^{(k)T})$, $\mathbf{r}_u^{(k)} = \sum_i([\mathbf{S}]_{u,i} \boldsymbol{\psi}_i^{(k)})$, (k) is the iteration index for SGD, and $\widehat{\nabla f_1(\boldsymbol{\theta}_u)}$ is the plain SGD over one random sample $u \in \mathcal{U}_i$. Similarly, the update for the sub-problem (S2) is done by taking T plain SGD steps of $f_2(\boldsymbol{\psi}) = \sum_{i \in \mathcal{I}_u} h_i(\boldsymbol{\psi}_i)$, i.e., the SGD, $\widehat{\nabla f_2(\boldsymbol{\psi}_i)} = \widehat{\nabla}(\sum_{i \in \mathcal{I}_u} h_i(\boldsymbol{\psi}_i)) = h_i(\boldsymbol{\psi}_i)$, where i is single random index from \mathcal{I}_u . Thus, the SGD MF update for the sub-problem (S2) is expressed as

$$\begin{aligned} \boldsymbol{\psi}_i^{(k+1)} &= \boldsymbol{\psi}_i^{(k)} - \alpha_k \widehat{\nabla f_2(\boldsymbol{\psi}_i^{(k)})} = \boldsymbol{\psi}_i^{(k)} - \alpha_k \nabla h_2(\boldsymbol{\psi}_i^{(k)}), \quad i \in \mathcal{I}_u \\ &= \boldsymbol{\psi}_i^{(k)} + 2\alpha_k ((\sum_u([\mathbf{S}]_{u,i} \boldsymbol{\theta}_u^{(k)T})) - (\sum_u(\boldsymbol{\theta}_u^{(k)} \boldsymbol{\theta}_u^{(k)T})) + \lambda_i \mathbf{I}_D) \boldsymbol{\psi}_i^{(k)}, \quad i \in \mathcal{I}_u, \forall k = 1..T \end{aligned}$$

where i is a single index chosen randomly from \mathcal{I}_u , $\mathbf{t}_i^{(k)} = \sum_u([\mathbf{S}]_{u,i} \boldsymbol{\theta}_u^{(k)T})$, $\mathbf{P}_i^{(k)} = \sum_u(\boldsymbol{\theta}_u^{(k)} \boldsymbol{\theta}_u^{(k)T})$, and $\widehat{\nabla f_2(\boldsymbol{\psi}_i)}$ is the plain SGD gradient computed with one sample $i \in \mathcal{I}_u$, chosen at random. We write the SGD MF updates as

$$\begin{cases} \boldsymbol{\theta}_u^{(k+1)} &= \boldsymbol{\theta}_u^{(k)} + 2\alpha_k ((\sum_i([\mathbf{S}]_{u,i} \boldsymbol{\psi}_i^{(k)})) - ((\sum_i \boldsymbol{\psi}_i^{(k)} \boldsymbol{\psi}_i^{(k)T}) + \mu_u \mathbf{I}_D) \boldsymbol{\theta}_u^{(k)}), \quad u \in \mathcal{U}_i \\ \boldsymbol{\psi}_i^{(k+1)} &= \boldsymbol{\psi}_i^{(k)} + 2\alpha_k ((\sum_u([\mathbf{S}]_{u,i} \boldsymbol{\theta}_u^{(k)T})) - (\sum_u(\boldsymbol{\theta}_u^{(k)} \boldsymbol{\theta}_u^{(k)T})) + \lambda_i \mathbf{I}_D) \boldsymbol{\psi}_i^{(k)}, \quad i \in \mathcal{I}_u \end{cases} \quad \forall k = 0, 1, \dots, T, \tag{9}$$

where u is a random index chosen from \mathcal{U}_i , and i a random index from \mathcal{I}_u . $0 \leq \alpha_k \leq 1$ is the step size for SGD.

BGD for MF (BGD MF): Rather than having a closed-form solution for each optimization block, BGD proceeds by taking T gradient steps for each block T gradient step. We skip the details here for space limitations. Thus, the BGD updates for the MF problem are expressed as

$$\begin{cases} \boldsymbol{\theta}_u^{(k+1)} &= \boldsymbol{\theta}_u^{(k)} + 2\alpha_k ((\sum_i([\mathbf{S}]_{u,i} \boldsymbol{\psi}_i^{(k)})) - ((\sum_i \boldsymbol{\psi}_i^{(k)} \boldsymbol{\psi}_i^{(k)T}) + \mu_u \mathbf{I}_D) \boldsymbol{\theta}_u^{(k)}) \\ \boldsymbol{\psi}_i^{(k+1)} &= \boldsymbol{\psi}_i^{(k)} + 2\alpha_k ((\sum_u([\mathbf{S}]_{u,i} \boldsymbol{\theta}_u^{(k)T})) - (\sum_u(\boldsymbol{\theta}_u^{(k)} \boldsymbol{\theta}_u^{(k)T})) + \lambda_i \mathbf{I}_D) \boldsymbol{\psi}_i^{(k)} \end{cases}$$

$$\forall (u, i) \in \mathcal{K}, k = 0, 1, \dots, T, \tag{10}$$

where (u, i) are the codebook indexes at UE and BS, k is the GD iteration index, and $\alpha^{(k)}$ is the BGD step size ($0 < \alpha^{(k)} < 1$).

4.3. Solutions for NMF

Our proposed NMF follows the exact steps as in MF, with the main difference of constraining the latent vectors being non-negative $\theta_u \in \mathbb{R}_+^D, \psi_i \in \mathbb{R}_+^D, \forall (u, i) \in \mathcal{K}$. Likewise, we solve the NMF problem, (P2), using BCD, SGD, and BGD.

BCD for NMF (BCD NMF): The derivations of BCD for NMF (11) are identical to those of BCD for MF (8), followed by the corresponding projection operation. The updates of BCD for NMF derivations are given by

$$\begin{cases} \theta_u^{(k+1)} &= \left[(\sum_i \psi_i^{(k)} (\psi_i^{(k)})^T) + \mu_u \mathbf{I} \right]^{-1} (\sum_i [\mathbf{S}]_{u,i} \psi_i^{(k)}) \Big|_+ \\ \psi_i^{(k+1)} &= \left[(\sum_u \theta_u^{(k+1)} (\theta_u^{(k+1)})^T) + \lambda_i \mathbf{I} \right]^{-1} (\sum_u [\mathbf{S}]_{u,i} \theta_u^{(k+1)}) \Big|_+ \end{cases} \tag{11}$$

$\forall (u, i) \in \mathcal{K}, k = 0, 1, \dots, I_M$

where $^{(k)}$ is the BCD iteration index, and $[\mathbf{a}]_+ := \max(\mathbf{a}, \mathbf{0})$ is applied element by element on \mathbf{a} , i.e., a Euclidean projection of \mathbf{a} on \mathbb{R}_+^D . Since the projection is Euclidean (non-expansive operator), the corollary stated in the previous subsection applies to the BCD for NMF too.

Block Stochastic Gradient Descent (BSGD) for NMF (SGD NMF): The SGD NMF derivations are exactly the same as that of SGD MF, followed by a projection $[\]_+$. We thus express the SGD NMF updates as

$$\begin{cases} \theta_u^{(k+1)} &= \left[\theta_u^{(k)} + 2\alpha_k ((\sum_i ([\mathbf{S}]_{u,i} \psi_i^{(k)})) - ((\sum_i \psi_i^{(k)} \psi_i^{(k)T}) + \mu_u \mathbf{I}_D) \theta_u^{(k)}) \right]_+, u \in \mathcal{U}_i \\ \psi_i^{(k+1)} &= \left[\psi_i^{(k)} + 2\alpha_k ((\sum_u ([\mathbf{S}]_{u,i} \theta_u^{(k)T})) - (\sum_u (\theta_u^{(k)} \theta_u^{(k)T})) + \lambda_i \mathbf{I}_D) \psi_i^{(k)} \right]_+ \end{cases} \tag{12}$$

$\forall k = 0, 1, \dots, T,$

where u is a random index chosen from \mathcal{U}_i, i is a random index from $\mathcal{I}_u, [\mathbf{a}]_+ := \max(\mathbf{a}, \mathbf{0})$, and $\alpha^{(k)}$ is the SGD step size ($0 < \alpha^{(k)} < 1$).

BGD for NMF (BGD NMF): The solution and derivations for BGD NMF are the same as those for BGD MF, followed by a projection $[\]_+$, i.e,

$$\begin{cases} \theta_u^{(k+1)} &= \left[\theta_u^{(k)} + 2\alpha_k ((\sum_i ([\mathbf{S}]_{u,i} \psi_i^{(k)})) - ((\sum_i \psi_i^{(k)} \psi_i^{(k)T}) + \mu_u \mathbf{I}_D) \theta_u^{(k)}) \right]_+ \\ \psi_i^{(k+1)} &= \left[\psi_i^{(k)} + 2\alpha_k ((\sum_u ([\mathbf{S}]_{u,i} \theta_u^{(k)T})) - (\sum_u (\theta_u^{(k)} \theta_u^{(k)T})) + \lambda_i \mathbf{I}_D) \psi_i^{(k)} \right]_+ \end{cases} \tag{13}$$

$\forall (u, i) \in \mathcal{K}, k = 0, 1, \dots, T,$

where $[\mathbf{a}]_+ := \max(\mathbf{a}, \mathbf{0}), ^{(k)}$ is the GD iteration index and $\alpha^{(k)}$ is the GD step size ($0 < \alpha^{(k)} < 1$). We use a constant step size $\alpha_k = \alpha$ for all these methods.

4.4. Prediction for MF and NMF

For both MF and NMF, the predicted RSE of the beam-pair (u, i) , for beam indexes that were not sounded, is expressed as

$$\{ \widehat{RSE}_{u,i} := (\widehat{\theta}_u)^T \widehat{\psi}_i \mid \forall (u, i) \in \mathcal{L} \} \tag{14}$$

where \mathcal{L} is the test set and $\{\widehat{\theta}_u, \widehat{\psi}_i\}$ are optimal solutions to MF (or NMF). Afterwards, we search for the optimal beam pair at UE and BS as the one with the highest RSE value over both training and test sets, as follows:

$$(u^*, i^*) = \operatorname{argmax}_{(u,i) \in \mathcal{L} \cup \mathcal{K}} (\widehat{\theta}_u)^T \widehat{\psi}_i. \tag{15}$$

4.5. Proposed BA Algorithm Using MF/NMF

Due to the fact that the updates given in a closed-form solution, we can quantify the computational complexity of all of the above methods. As seen from the updates for BCD MF and BCD NMF, we have to invert two $D \times D$ matrices (for the sum problems S1 and S2). Thus, the (per-iteration) computational complexity of BCD MF and BCD NMF is approximated as $C_{BCD MF} = C_{BCD NMF} = \mathcal{O}(2D^3)$. Moreover, for BGD MF and BGD NMF, one has to compute two full-batch gradients over all training samples in \mathcal{K} (for the sub-problems S1 and S2). Consequently, the complexity, per-iteration, for BGD MF and BGD NMF is approximated as $C_{BGD MF} = C_{BGD NMF} = \mathcal{O}(2|\mathcal{K}|)$. Finally, for SGD MF and SGD NMF, since we use a mini-batch size = 1 (for the sub-problems S1 and S2), the resulting per-iteration computational complexity is approximated as $C_{SGD MF} = C_{SGD NMF} = \mathcal{O}(2)$. Solving the MF and NMF problem, we employ methods such as BCD, BGD, or SGD. All details are shown in Algorithm 1.

Algorithm 1 Proposed MF/NMF-Based BA Method.

Input: $\{\mathbf{f}_u\}_{\forall u \in \mathcal{T}}, \{\mathbf{W}_i\}_{\forall i \in \mathcal{R}}, \eta, P_u$
 - Generate randomly sub-sampled codebooks, $\mathcal{T}_S, \mathcal{R}_S$, satisfying $(|\mathcal{T}_S| \cdot |\mathcal{R}_S|) / (|\mathcal{T}| \times |\mathcal{R}|) = \eta$
 - Sound beam pairs from training set, $\mathcal{K} := \mathcal{T}_S \times \mathcal{R}_S$.
 - Record corresponding RSE in and generate mat. \mathbf{S} , in (5)
 - Select model: MF or NMF
 - **IF** MF model selected
 solve (P1) with BCD for MF, in (8) or solve (P1) with BGD for MF, in (10) or solve (P1) with SGD for MF, in (9). At the end of training, return optimal latent vectors, $\{\hat{\boldsymbol{\theta}}_u, \hat{\boldsymbol{\psi}}_i\}_{(u,i) \in \mathcal{K}}$
 - **IF** NMF model selected
 solve (P2) with BCD for NMF, in (11) or solve (P2) with BGD for NMF, in (13) or solve (P2) with SGD for NMF, in (12). At the end of training, return ideal latent vectors, $\{\hat{\boldsymbol{\theta}}_u, \hat{\boldsymbol{\psi}}_i\}_{(u,i) \in \mathcal{K}}$
 - Use ideal latent vectors $\{\hat{\boldsymbol{\theta}}_u, \hat{\boldsymbol{\psi}}_i\}_{(u,i) \in \mathcal{K}}$, to predict unknown RSE of test set, \mathcal{L} , in (14)
 - Search training and test sets, for beam pair w/ largest RSE, (15)
 Output: $\mathbf{f}_{u^*}, \mathbf{W}_{i^*}$

While, for MF BCD and NMF BCD, the only hyperparameter is the model size D , MF BGD and NMF BGD require, in addition to D, α^k , the GD step size as hyperparameters.

4.6. Numerical Simulations

This section illustrates our numerical setup. The number of antennas at UE and BS $\in \{128, 256, 512, 1024\}$. We set up $N_T = C_T$ and $N_R = C_R$. The overhead ratio regime $\eta \in \{0.7, 0.5, 0.3, 0.1\}$. The number of OFDM sub-carriers $N_c = 64$ and the number of channel paths L is 2. We vary the transmitted power, $P_u \in \{1, 10^{-1}, 10^{-2}\}$. We use DFT codebooks at UE and BS. The optimal hyperparameters are chosen to minimize test loss. The model dimension $D \in \{2, 3, 4, 5, 6\}$, the learning rate $\alpha_k \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$, and the regularization factors $\{\lambda, \mu\} \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$. For each MIMO configuration and for each P_u regime, we randomly generate and store the resulting RSE matrices.

We propose to investigating six models in total (BCD MF, BCD NMF, BGD MF, BGD NMF, SGD MF, SGD NMF) with respect to three transmitted power regimes: high $P_u = 1W$, medium $P_u = 10^{-1} W$, and low $P_u = 10^{-2} W$ with fixed $\sigma^2 = 1$. In Table 1, we provide a summary for all proposed system parameters. We use the training Normalized MSE (NMSE) to evaluate the training error, expressed as Train NMSE = $\frac{1}{|\mathcal{K}|} (\sum_{(u,i) \in \mathcal{K}} (\frac{\hat{\boldsymbol{\theta}}_u^T \hat{\boldsymbol{\psi}}_i - RSE_{u,i}}{RSE_{u,i}})^2)$. We also define Test NMSE = $\frac{1}{|\mathcal{L}|} (\sum_{(u,i) \in \mathcal{L}} (\frac{\widehat{RSE}_{u,i} - \hat{\boldsymbol{\theta}}_u^T \hat{\boldsymbol{\psi}}_i}{\widehat{RSE}_{u,i}})^2)$. The range of training error and the overall behavior of BCD-based models are different and distinctive from GD models in both MF and NMF; for instance, BGD-based models' error range are around $\times 10^{-7}$, while BCD-based models are around $\times 10^{-4}$. Thus, GD is more accurate. However, BCD converges faster and the cost function drops to low values from the very first iterations. In addition, for MF and NMF, the train NMSE decreases with

the increase in the overhead ratio η , as seen in Figure 5. Low and medium P_u regimes are characterized by noisy links between *UE* and *BS* and represent a more challenging experimental environment. *BCD*-based models tend to be faster in reaching low error values, while *BGD*-based models are more accurate. (For instance, *BSGD* generally ameliorates the quality of prediction compared with *BGD*).

Table 1. System parameters and hyperparameters.

System Configuration for All Proposed Models	
System parameter	Numerical value
number of antennas N_T at <i>UE</i>	128, 256, 512, 1024
number of antennas N_R at <i>BS</i>	128, 256, 512, 1024
codebook cardinality $ \mathcal{T} $ at <i>UE</i>	128, 256, 512, 1024
codebook cardinality $ \mathcal{R} $ at <i>BS</i>	128, 256, 512, 1024
overhead ratio η regime	0.7, 0.5, 0.3, 0.1
number of <i>OFMD</i> sub-carriers N_c	64
number of channel paths L	2 (NLoS)
transmitted power P_u (W)	$1, 10^{-1}, 10^{-2}$
<i>MF/NMF</i> dimension D_{MF}	2, 3, 4, 5, 6
<i>MF/NMF</i> learning rate α_k	$10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$
<i>MF/NMF</i> regularization factors λ, μ	$10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}$
<i>MLP</i> number of layers J	1, 2, 3
<i>MLP</i> number of neurons per layer D_{MLP}	8, 16, 32, 64, 128
<i>MLP</i> batch size B	2, 4, 8, 16, 32, 64, 128
<i>MLP</i> learning rate β_k	$10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$

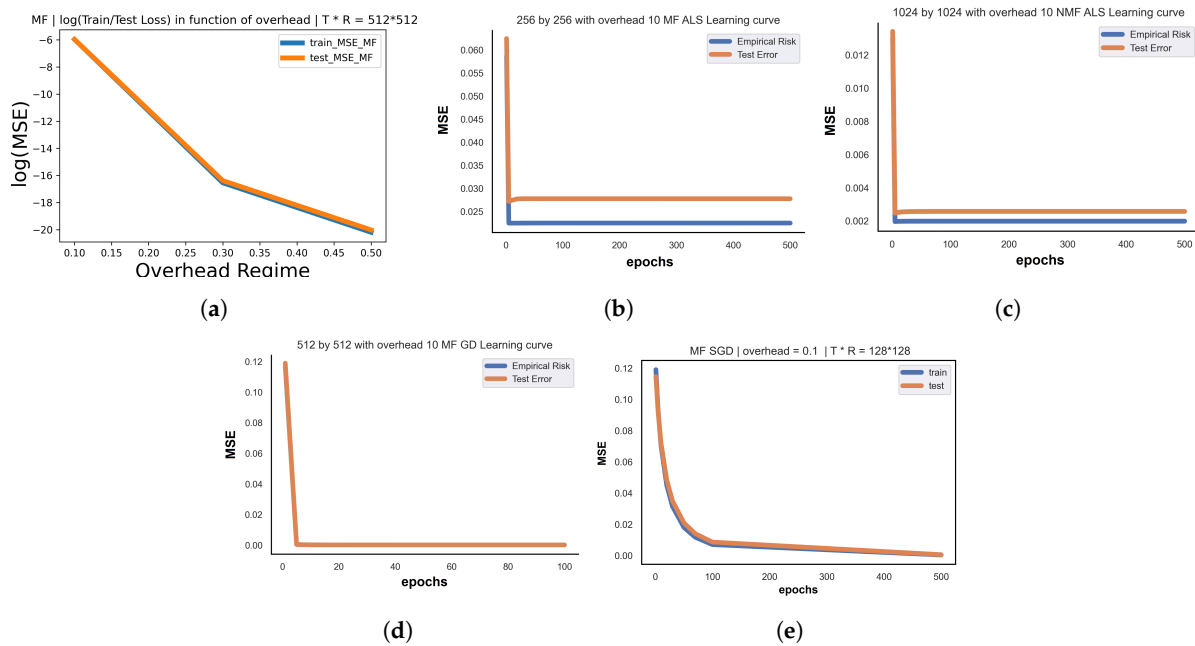


Figure 5. *MF/NMF* train/test performance and learning curves: (a) 512×512 train/test loss in function of the overhead ratio; (b) learning curve: 256×256 with overhead 0.1 *BCDMF*; (c) learning curve: 1024×1024 with overhead 0.1 *BCDNMF*; (d) learning curve: 512×512 with overhead 0.1 *BGD MF*; (e) learning curve: 128×128 with overhead 0.1 *BCDSGD*.

Regarding *MF/NMF* simulation figures, Figure 5a states the decrease of train/test *NMSE* in function of the overhead ratio (more training samples result in fewer errors); Figure 5b,c track the instant drop in loss values from the very first iterations for *BCD*-based models; and Figure 5d,e present the progressive convergence of cost function among the iterations when we use *BGD*-based models. In summary, Table 2 outlines the optimal (minimum) signaling overhead ratio required for the all proposed system configurations, the optimal model (holding the smallest total cost function), the related combination of optimal hyperparameters, and the corresponding train/test error values. When the signal is affected with much noise, it is harder to keep the same range of error when compared with high a P_u regime. In fact, *MF* models keep the same (minimum) signaling overhead (0.1) regardless of the transmitted power regime, being able to accurately predict with just 10% of sounded beams. Thus, the proposed *MF/NMF* methods are able to reduce the pilot signaling overhead by 90% compared with Exhaustive *BA* with negligible training and test errors.

Table 2. *QoS* minimum overhead required for *MF/NMF* for all proposed P_u regimes.

a <i>MF/NMF</i> <i>QoS</i> Minimum Overhead Required for $P_u = 1$ W				
MIMO setup	Optimal hyperparameters	Min Overhead	Train NMSE	Test NMSE
128 by 128	BGD NMF{D = 2, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 0.001$ }	0.1	8.407746×10^{-6}	9.147875×10^{-6}
256 by 256	BGD MF{D = 3, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 0.001$ }	0.1	4.102708×10^{-6}	7.344720×10^{-6}
512 by 512	BGD MF{D = 4, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 0.001$ }	0.1	8.374633×10^{-7}	9.417057×10^{-7}
1024 by 1024	SGD NMF{D = 4, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 0.01$ }	0.1	1.219227×10^{-7}	1.616363×10^{-7}
b <i>MF/NMF</i> <i>QoS</i> Minimum Overhead Required for $P_u = 10^{-1}$ W				
MIMO setup	Optimal hyperparameters	Min Overhead	Train NMSE	Test NMSE
128 by 128	SGD NMF {D = 2, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 0.001$ }	0.1	0.000191	0.000276
256 by 256	SGD NMF {D = 3, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 0.001$ }	0.1	4.648861×10^{-5}	5.775554×10^{-5}
512 by 512	BGD NMF{D = 4, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 0.001$ }	0.1	1.052556×10^{-5}	1.170430×10^{-5}
1024 by 1024	BGD NMF {D = 4, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 0.001$ }	0.1	1.600790×10^{-6}	1.695907×10^{-6}
c <i>MF/NMF</i> <i>QoS</i> Minimum Overhead Required for $P_u = 10^{-2}$ W				
MIMO setup	Optimal hyperparameters	Min overhead	Train NMSE	Test NMSE
128 by 128	SGD MF {D = 2, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 1 \times 10^{-6}$ }	0.1	0.115517	0.118776
256 by 256	BGD MF {D = 3, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 0.0001$ }	0.1	0.016475	0.016679
512 by 512	SGD NMF{D = 4, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 1 \times 10^{-6}$ }	0.1	0.003371	0.003449
1024 by 1024	BGD MF {D = 4, $(\lambda, \mu) = (0.0001, 0.0001)$, $\alpha_k = 1 \times 10^{-5}$ }	0.1	0.001681	0.001948

5. Multi-Layer Perceptron

5.1. MLP Problem Formulation

We consider a feed-forward *MLP*, with J layers, modeled as a composition of J non-linear functions/layers. Let $z_0 \in \mathbb{R}$ be the *MLP* input, and $z_j \in \mathbb{R}$ be the *MLP* output; see Figure 6. We denote with $\{z_2, \dots, z_{J-1}\}$ all the hidden layers. We assume for simplicity that the width of all the layers is the same, denoted as D , i.e., $\{z_2 \in \mathbb{R}^D, \dots, z_{J-1} \in \mathbb{R}^D\}$; see Figure 6. The equation describing layer 1 is given by $z_1 = \sigma_1(\phi_1 z_0) = \sigma_1(\phi_1 1)$, where $z_1 \in \mathbb{R}^D$ is the output of layer 1, $\phi_1 \in \mathbb{R}^D$ is the resulting weight vector, and $\sigma_1(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^D$ is the non-linear activation function for layer 1. We use one hot encoding for the *MLP* input $z_0 \in \mathbb{R}$, i.e., $z_0 = 1$ for all training samples, $\forall (u, i) \in \mathcal{K}$. We express the output of the hidden layers, $\{z_j \in \mathbb{R}^D\}_{j=2}^{J-1}$, as $z_j = \sigma_j(\Phi_j z_{j-1})$, $\forall j \in \{2, \dots, J-1\}$, where $z_{j-1} \in \mathbb{R}^D$ is the input of the layer j and $z_j \in \mathbb{R}^D$ is its output, $\forall j \in \{2, \dots, J-1\}$; $\Phi_j \in \mathbb{R}^{D \times D}$ is the weight matrix for the layer j , $\forall j \in \{2, \dots, J-1\}$; and $\sigma_{j-1}(\cdot) : \mathbb{R}^D \rightarrow \mathbb{R}^D$ is the

element-by-element non-linear activation function for the layer j , $\forall j \in \{2, \dots, J - 1\}$. Finally, the relation for the last layer $j = J$ is expressed as $z_J = \sigma_J(\phi_J z_{J-1})$, where $z_J \in \mathbb{R}$ is the output for layer J , $\phi_J \in \mathbb{R}^{1 \times D}$ is its weight vector, and $\sigma_j() : \mathbb{R}^D \rightarrow \mathbb{R}$ is the non-linear activation function for the layer J . We express the output of the MLP $z_J \in \mathbb{R}$ (as a function of all layers) as

$$z_J := \sigma_J(\phi_J \dots \sigma_2(\Phi_2(\sigma_1(\phi_1)))) \tag{16}$$

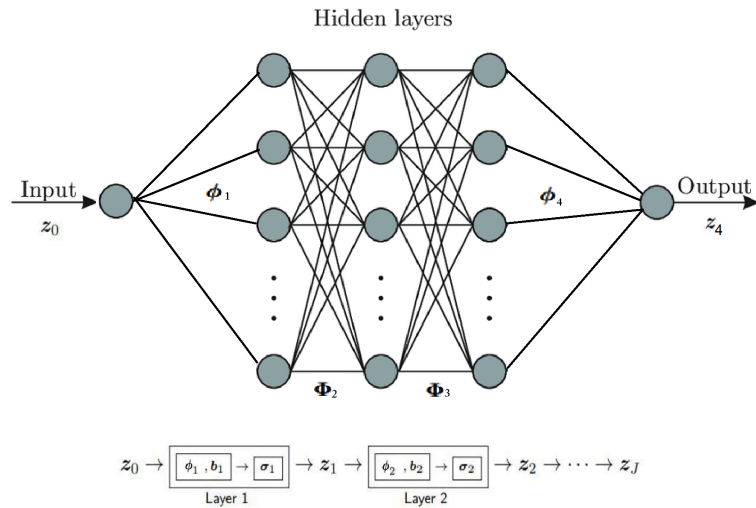


Figure 6. Multi-Layer Perceptron architecture (toy example with $J = 4$).

The output of *MLP* is made to fit/approximate all the *RSE* values at all training samples; $z_J := RSE_{u,i}, \forall (u, i) \in \mathcal{K}$. We define the MSE loss $l_{u,i}$ for the sample (u, i) in the training set \mathcal{K} as the distance between the MLP output z_J and the known *RSE* label for the beam pair (u, i) , $RSE_{u,i}$, i.e.,

$$l_{u,i} := (z_J - RSE_{u,i})^2 = \left(\underbrace{\sigma_J(\phi_J \dots \sigma_2(\Phi_2(\sigma_1(\phi_1))))}_{\text{MLP output}} - \underbrace{RSE_{u,i}}_{\text{RSE value}} \right)^2, \forall (u, i) \in \mathcal{K}$$

Then, the empirical risk is defined as the average of the individual loss $l_{u,i}$ across the training set \mathcal{K} , $(1/|\mathcal{K}|) \sum_{(u,i) \in \mathcal{K}} l_{u,i}$. The empirical risk minimization for the MLP is given in (P3).

$$(P3) := \{(\phi_1^*, \Phi_2^*, \dots, \phi_J^*) \left\{ \begin{array}{l} \text{argmin}_{\phi_1, \Phi_2, \dots, \phi_J} \frac{1}{|\mathcal{K}|} \sum_{(u,i) \in \mathcal{K}} l_{u,i}(\phi_1, \Phi_2, \dots, \Phi_{J-1}, \phi_J) \\ \text{s. t. } \phi_1 \in \mathbb{R}^D, \Phi_2 \in \mathbb{R}^{D \times D}, \dots, \Phi_{J-1} \in \mathbb{R}^{D \times D}, \phi_J \in \mathbb{R}^{1 \times D} \end{array} \right.$$

5.2. MLP Learning

We propose to learn the optimal *MLP* weights via back-propagation (BP). We choose an arbitrary mini-batch of samples of size $\mathcal{B} \subseteq \mathcal{K}$ and define the mini-batch loss as

$$l_{\mathcal{B}} := \frac{1}{|\mathcal{B}|} \sum_{(u,i) \in \mathcal{B}} (\sigma_J(\phi_J \dots \sigma_2(\Phi_2(\sigma_1(\phi_1)))) - RSE_{u,i})^2, \forall (u, i) \in \mathcal{B} \tag{17}$$

We express the partial derivative of the loss corresponding to the mini-batch $l_{\mathcal{B}}$ with respect to each layer $\Phi_j, j \in \{1, \dots, J\}$ as

$$\frac{\partial l_{\mathcal{B}}}{\partial \Phi_j} = \frac{1}{|\mathcal{B}|} \sum_{(u,i) \in \mathcal{B}} (\delta_j z_{j-1}^T), \forall j \in \{1, \dots, J\}, \tag{18}$$

where

$$\delta_j \triangleq \begin{cases} (\Phi_{j+1}^T \delta_{j+1}) \circ \sigma_j', j < J \\ 2(z_j - RSE_{u,i}) \circ \sigma_j', j = J, (u, i) \in \mathcal{B} \end{cases}, \sigma_j' \triangleq \frac{\partial \sigma(u)}{\partial u} = \left[\frac{\partial \sigma(u_1)}{\partial u_1}, \dots, \frac{\partial \sigma(u_{d_j})}{\partial u_{d_j}} \right]^T,$$

$j = 1, \dots, J$ and \circ denotes the Hadamard product. We express the BP weight update of the mini-batch loss l_B , for all layers $\forall j \in \{1, \dots, J\}$, as

$$\Phi_j^{(k+1)} = \Phi_j^{(k)} - \beta_j^{(k)} \frac{\partial l_B}{\partial \Phi_j} \Big|_{\Phi_j^{(k)}}, \forall j \in \{1, \dots, J\}, k = \{1, \dots, T\} \tag{19}$$

where $^{(k)}$ is the BP iteration index, $\Phi_j^{(k)}$ is the value of Φ_j at iteration k , $\beta_j^{(k)}$ is the BP step size (learning rate) for the layer j at iteration k , and $\frac{\partial l_B}{\partial \Phi_j} \Big|_{\Phi_j^{(k)}}$ is the partial derivative given in (18) evaluated at $\Phi_j^{(k)}$.

Back-propagation algorithm with mini-batch

Choose the mini-batch \mathcal{B} as a random subset of the training set \mathcal{K} .

1. Compute the loss function l_B for all samples in the mini-batch $(u, i) \in \mathcal{B}$ in (17).
2. Compute the partial derivative $\frac{\partial l_B}{\partial \Phi_j}$ of the mini-batch loss l_B with respect to Φ_j in (18).
3. Update the weights of each layer as in (19).

We assume that the BP learning rate is the same for all layers, $\beta_j^{(k)} = \beta^k, \forall j \in \{1, \dots, J\}$.

5.3. Prediction Using MLP

The MLP prediction for the sample (u, i) in the test set \mathcal{L} , using optimal weights ϕ_1^* , $\Phi_2^*, \dots, \phi_J^*$ is as follows:

$$\hat{z}_J = \sigma_J(\phi_J^* \dots \sigma_2(\Phi_2^*(\sigma_1(\phi_1^*))))), \forall (u, i) \in \mathcal{L} \tag{20}$$

Therefore, the test MSE is defined as

$$\frac{1}{|\mathcal{L}|} \sum_{(u,i) \in \mathcal{L}} \left(\widehat{RSE}_{u,i} - \sigma_J(\phi_J^* \dots \sigma_2(\Phi_2^*(\sigma_1(\phi_1^*)))) \right)^2 \tag{21}$$

We then select the optimal indexes u^* and i^* related to the highest $RSE_{u,i}$ value, as follows:

$$(u^*, i^*) = \operatorname{argmax}_{(u,i) \in \mathcal{L} \cup \mathcal{K}} \{RSE_{u,i} | \forall (u, i) \in \mathcal{K}\} \cup \{R\hat{S}E_{u,i} | \forall (u, i) \in \mathcal{L}\} \tag{22}$$

5.4. Proposed BA Algorithm Using MLP

The Multi-Layer Perceptron-based Beam Alignment is specified in Algorithm 2.

Algorithm 2 Proposed MLP-Based BA Method.

- Input: $\{f_u\}_{\forall u \in \mathcal{T}}, \{W_i\}_{\forall i \in \mathcal{R}}, \eta, Pu$
- Generate randomly sub-sampled codebooks, $\mathcal{T}_S, \mathcal{R}_S$, satisfying $(|\mathcal{T}_S| \cdot |\mathcal{R}_S|) / (|\mathcal{T}| \times |\mathcal{R}|) = \eta$
 - Sound beam pairs from training set, $\mathcal{K} := \mathcal{T}_S \times \mathcal{R}_S$.
 - Record corresponding RSE and generate RSE mat. \mathbf{S} , in (5)
 - Train MLP weights (using back-propagation algorithm)
 return optimal weights, $\{\phi_1^*, \Phi_2^*, \dots, \phi_J^*\}$
 - Use optimal parameters $\{\phi_1^*, \Phi_2^*, \dots, \phi_J^*\}$, to predict unknown RSE of test set, \mathcal{L} , in (21)
 - Search training and test sets, for optimal beam pair (u^*, i^*) , holding the largest RSE, (22)
- Output: f_{u^*}, W_{i^*}
-

We assume that the number of neurons per layer D , the number of layers J , the mini-batch size $B = |\mathcal{B}|$, and the BP learning rate $\beta^{(k)}$ are hyperparameters. They are tuned using a grid search cross-validation.

5.5. Numerical Simulations

We define the training and test cost functions as follows:

$$\text{Train NMSE} = \frac{1}{|\mathcal{K}|} \left(\sum_{(u,i) \in \mathcal{K}} \left(\frac{RSE_{u,i} - \sigma_J(\phi_J \dots \sigma_2(\Phi_2(\sigma_1(\phi_1))))}{RSE_{u,i}} \right)^2 \right) \quad (23)$$

$$\text{Test NMSE} = \frac{1}{|\mathcal{L}|} \left(\sum_{(u,i) \in \mathcal{L}} \left(\frac{\widehat{RSE}_{u,i} - \sigma_J(\phi_J^* \dots \sigma_2(\Phi_2^*(\sigma_1(\phi_1^*))))}{\widehat{RSE}_{u,i}} \right)^2 \right) \quad (24)$$

Therefore, we used the same system configurations as for *MF/NMF*, resumed in Table 1. Moreover, we choose the learning rate $\beta_k \in \{0.1, 0.01, 0.001, 0.0001\}$, while the batch size $B \in \{2, 4, 8, 16, 32, 64, 128\}$, the number of hidden layers $J \in \{1, 2, 3\}$. For each layer, the number of neurons $D \in \{8, 16, 32, 64, 128\}$. We use the Rectified Linear Units as our activation function for all layers.

Similar to *MF/NMF*, train performance is observed when we track the evolution of the cost function *NMSE*, applied to the training samples of the set \mathcal{K} , in a function of iterations. The range of considerably low-error values and the overall learning behavior of the *MLP* architecture illustrates that our shallow neural network successfully resolves the non-linear regression problems related to our BA process. For massive setups, *MLP* reaches around 10^{-6} error in a high P_u regime. However, this cost value increases as long as the amount of noise and interference augments. Note that the train *NMSE* also decreases when we increase the size of the dataset matrix \mathcal{S} , which provides more samples for *MLP* to improve the feature extraction and the prediction quality. Regarding the unknown beams, test error values in the numerical result tables are close to the train cost (with no overfitting or underfitting in the corresponding learning curves). Moreover, the test loss is impacted by the transmitted power regime the same way as the training process. Identical to *GD*-based *MF/NMF*, the *MLP* learning curves in Figure 7 plot the same shape of curve with a continuous monotonic decrease in the train and test cost among the iterations: the convergence is progressive among the iterations, and at the last epoch, training and test *NMSE* values land at considerably low error values and prove that *MLP* accurately fits to our problem and provides a concrete solution for *ML*-based *BA*. From a *QoS* perspective, Table 3 resumes the smallest (optimal) signaling overhead required for a successful beam sounding based on reliable prediction quality. Similar to *MF/NMF*, for all the proposed transmitted power, *MLP* requires 10% of the total beam pairs to fulfill the *RSE* matrix.

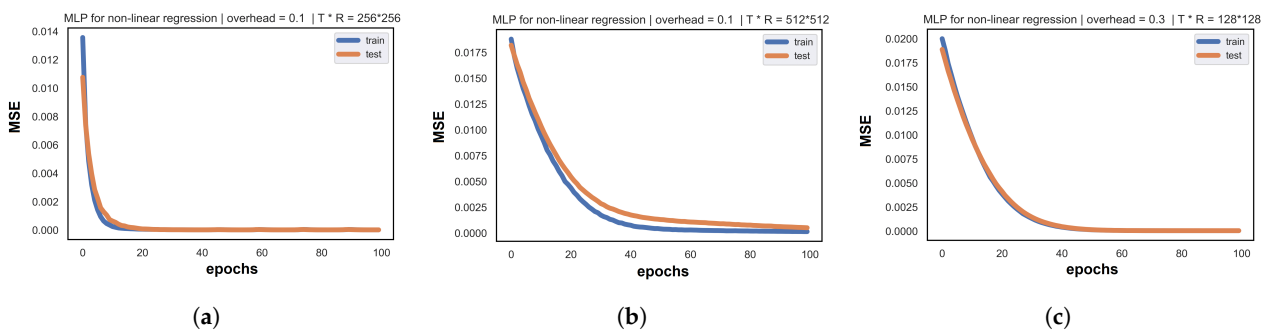


Figure 7. *MLP* Learning curves: (a) learning curve: 256×256 with overhead 0.1 *MLP*; (b) learning curve: 512×512 with overhead 0.1 *MLP*; and (c) learning curve: 128×128 with overhead 0.3 *MLP*.

Table 3. QoS minimum overhead required for MLP for all the proposed P_u regimes.

a MLP QoS Minimum Overhead Required for $P_u = 1$ W				
MIMO setup	Optimal hyperparameters	Min overhead	Train NMSE	Test NMSE
128 by 128	$\{J = 3, D = 8\}, B = 4, \beta_k = 0.0001$	0.1	0.001144	0.002639
256 by 256	$\{J = 3, D = 16\}, B = 16, \beta_k = 0.001$	0.1	3.941522×10^{-5}	3.948157×10^{-6}
512 by 512	$\{J = 3, D = 64\}, B = 32, \beta_k = 0.0001$	0.1	3.305507×10^{-5}	3.335168×10^{-5}
1024 by 1024	$\{J = 3, D = 64\}, B = 64, \beta_k = 0.0001$	0.1	9.810028×10^{-6}	9.857067×10^{-6}
b MLP QoS Minimum Overhead Required for $P_u = 10^{-1}$ W				
MIMO setup	Optimal hyperparameters	Min overhead	Train NMSE	Test NMSE
128 by 128	$\{J = 3, D = 8\}, B = 4, \beta_k = 0.0001$	0.1	0.007569	0.007662
256 by 256	$\{J = 3, D = 16\}, B = 16, \beta_k = 0.001$	0.1	0.000139	0.000288
512 by 512	$\{J = 3, D = 64\}, B = 32, \beta_k = 0.0001$	0.1	5.419598×10^{-5}	5.756302×10^{-5}
1024 by 1024	$\{J = 3, D = 64\}, B = 64, \beta_k = 0.0001$	0.1	1.184073×10^{-5}	1.72301×10^{-5}
c MLP QoS Minimum Overhead Required for $P_u = 10^{-2}$ W				
MIMO setup	Optimal hyperparameters	Min overhead	Train NMSE	Test NMSE
128 by 128	$\{J = 3, D = 8\}, B = 4, \beta_k = 0.0001$	0.1	0.049559	0.071185
256 by 256	$\{J = 3, D = 16\}, B = 16, \beta_k = 0.001$	0.1	0.017011	0.017634
512 by 512	$\{J = 3, D = 64\}, B = 32, \beta_k = 0.0001$	0.1	0.000141	0.000666
1024 by 1024	$\{J = 3, D = 64\}, B = 64, \beta_k = 0.0001$	0.1	1.700140×10^{-4}	1.702889×10^{-4}

6. Results and Discussion

6.1. Train/Test Prediction Performance Comparison

For the six MF-based models, we select the best one (minimum test error) to represent the MF family of methods in this section and compare it with MLP. When we analyze QoS (Tables 1 and 2), we notice that the transmitted power regime impacts the quality of prediction by reducing the overall loss. For MF/NMF, we observe that the loss damage is large. We jump from around 10^{-8} for massive configurations (256, 512, and 1024) to 10^{-4} for smaller setups. For MLP, we spot the increase in the overall loss when we decrease P_u . Thus, MLP seems to be the most robust architecture with respect to changing the transmitted power. Additionally, we empirically notice that the change in the P_u values does not impact the optimal hyperparameters selected from cross-validation. Furthermore, when we track the evolution of the training/test cost in the function of iterations, we observe balanced models with no signs of overfitting or underfitting. On the other hand, when the transmitted power decreases, MF/NMF tend to be the most impacted models in terms of train/test error, while the MLP error is robust.

On the other hand, from a QoS perspective, concerning the evolution of the optimal (minimum) required signaling overhead and what impact can the P_u regime have on the optimal required values, in reference to Tables 1 and 2, all the proposed models required just 10% of the total number of beam pairs at UE and BS for all antenna configurations from 128×128 to 1024×1024 for all the proposed P_u values. This proves that the transmitted power impacts the quality of prediction but not the number of beam pairs required for training. In fact, low P_u leads to damaging the signal quality and subsequently damages the quantity of useful information to be extracted from the datasets. Finally, the only cases where the P_u regime impacts the optimal overhead ratio is among the smallest configurations, for instance, the 16×16 setup where it seems normal for all learning models to demand more data to learn from (more hidden interactions between UE and BS as features to extract). These are the experimental situations where Exhaustive BA is technically feasible.

6.2. Similarities and Differences between Models

All models required just 10% of the beams for training for all the proposed massive setups. Moreover, all the proposed models are shallow neural architectures with few hidden layers for low-complexity constraints. Even among the largest configurations, the optimal dimensions of models picked from the cross-validation illustrate small networks with no need to require dense architectures. Furthermore, all models succeeded with the matrix completion task, and they all illustrate a monotonic decrease in loss values as long as we increase the MIMO setup. Additionally, *MF*-based models are the most accurate reaching loss values in the range 10^{-8} for massive setups in a high P_u regime, and their cross-validation illustrates smaller grid search where there are fewer hyperparameters to tune. However, they are the slowest models when applied to high-dimensional MIMO setups. On the other hand, *MLP* illustrates a good balance between run time (complexity) and loss values (prediction quality). It reaches around 10^{-4} and 10^{-5} loss for massive configurations. In addition, the *MLP* is the most robust model facing the changes in the P_u values. In Figure 8, for 512×512 , the figure illustrates the train/test *NMSE* in the function of each model and the corresponding transmitted power: in Figure 8a, for $P_u = 1W$, *MF* achieves its best performance, slightly better than *MLP* with the difference between achieved cost values at around 10^{-1} . In Figure 8b, when $P_u = 10^{-1}W$, *MF* still gets the best performance, marginally better than *MLP* with an *NMSE* value difference of around 10^{-1} . In Figure 8c, when $P_u = 10^{-2}$, *MF* noticeably gets impacted (overall loss around 10^{-3}) while *MLP* provides the best prediction performance: this suggests that when P_u is small, *MLP* is more robust than *MF/NMF*, which performs best in high P_u regime. Similarly, almost same remarks hold for Figure 9 when we simulate the 128×128 configuration: in Figure 9a, *MF* reaches considerably better performance compared with *MLP* with 10^{-4} . In Figure 9b, *MLP* kept the same range of error, which states again the robustness of the model while *MF* got severely impacted (10^{-3}) but still holds the best performance. In Figure 9c, when P_u is weak, *MF* illustrates the worst performance in all simulations. On the other hand, *MLP* got slightly impacted with an overall loss of 10^{-1} and reaches the best quality of prediction. In Figure 10, we investigate the highest configuration 1024×1024 . Similar conclusions for Figures 8 and 9 hold for this figure in terms of best model (*MF* for $P_u = 1W$, $P_u = 10^{-1}$ and *MLP* for $P_u = 10^{-2}$). In addition, we aim to investigate the overall impact of varying the transmitted power. Thus, we track the $\log(\text{NMSE})$ values while switching from one P_u regime to another: In Figure 10, in Figure 10a, for *MLP*, the curve gap from low/medium is $\log(\text{NMSE})_{\text{medium}} - \log(\text{NMSE})_{\text{low}} \approx -16 - (-12) \approx -4$. The gap in the medium/high regimes is almost negligible ($\log(\text{NMSE})_{\text{high}} - \log(\text{NMSE})_{\text{medium}} \approx -16 - (-16) \approx 0.5$). Finally, in Figure 10b, the *MF* gap is around $\log(\text{NMSE})_{\text{medium}} - \log(\text{NMSE})_{\text{low}} \approx -17 - (-9) \approx -8$ and $\log(\text{NMSE})_{\text{high}} - \log(\text{NMSE})_{\text{medium}} \approx -22 - (-17) \approx -5$: at each change of P_u , *MF* is considerably impacted. To sum up, the choice of the optimal model strongly depends on the available complexity and the given transmitted power P_u . In fact, *MF*, whether through *BCD* or *BGD* optimization, is the best model when the transmitted power is high ($P_u = 1W$). In this case, *BCDMF* converges faster but has higher complexity than *BGD*. However, *SGD* for *MF/NMF* are the slowest models to converge but show negligible complexity. On the other hand, if we aim to prioritize run time, *MLP* exhibits the fastest predictions with good prediction error. Finally, it is wise to opt for *MLP* if the system is to operate under various transmitted power regimes where *MLP* offers good prediction quality for every P_u value and the available complexity is medium.

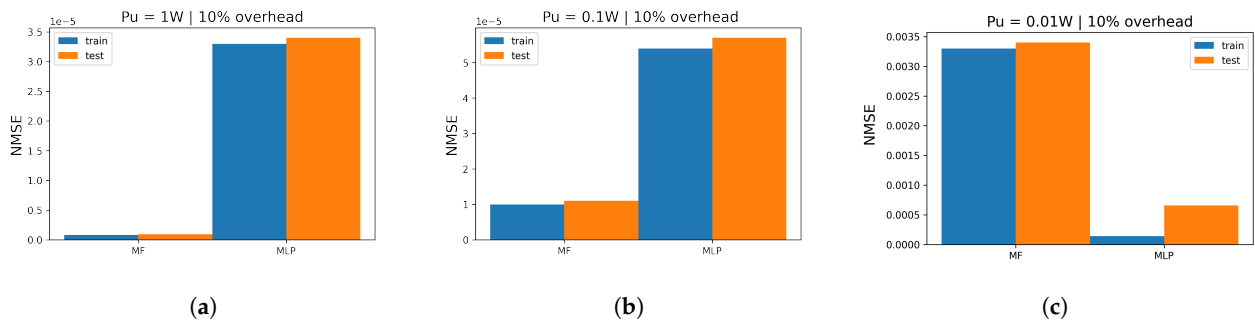


Figure 8. Train/test NMSE in function of P_u for all proposed models for 512×512 using optimal overhead ratio; (a) 512×512 train/test NMSE for $P_u = 1 W$; (b) 512×512 train/test NMSE for $P_u = 10^{-1} W$; (c) 512×512 train/test NMSE for $P_u = 10^{-2} W$.

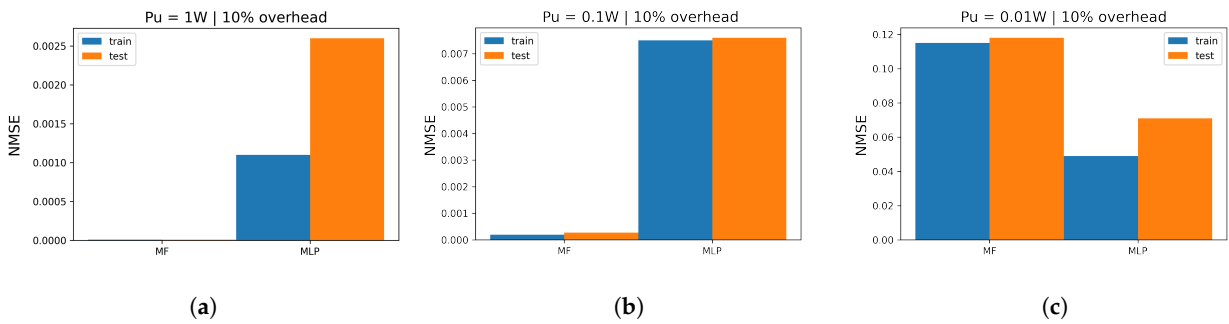


Figure 9. Train/test NMSE in function of P_u for all proposed models for 128×128 using optimal overhead ratio: (a) 128×128 train/test NMSE for $P_u = 1 W$; (b) 128×128 train/test NMSE for $P_u = 10^{-1}W$; (c) 128×128 train/test NMSE for $P_u = 10^{-2} W$.

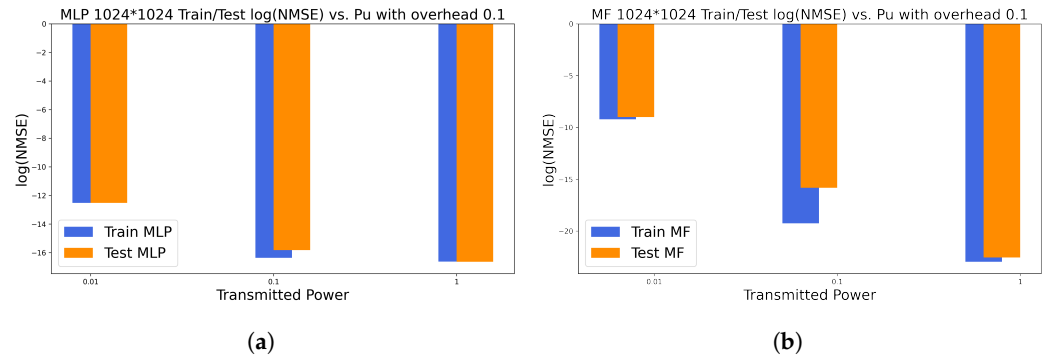


Figure 10. $\log(NMSE)$ in function of P_u for 1024×1024 using optimal overhead ratio: (a) MLP train/test $\log(NMSE)$ in function of P_u using optimal overhead ratio; (b) MF train/test $\log(NMSE)$ in function of P_u using optimal overhead ratio.

7. Conclusions

In this paper, we proposed a blind Machine Learning-based Beam Alignment using Matrix Factorization, non-negative Matrix Factorization, and Multi-Layer Perceptron. We assumed an Uplink massive mmWave MIMO system using single RF-chains at UE and multiple RF-chains at BS though a fully analog architecture. The proposed approach consists in sounding the RSE of sub-sampled codebooks at UE and BS. The RSE of the non-sounded beams is predicted using MF, NMF, and MLP models. Our results show that, by sounding just 10% of the total beam pair samples, we may predict with high accuracy the unknown RSE values, which massively reduce the large signaling overhead of Exhaustive BA. Our future work investigates the scalability of our approach to a multi-user scenario. Robustness and ML-interpretability are other research directions for modeling industrial deployment.

Author Contributions: Conceptualization, A.K., H.G. and G.R.-B.O.; Methodology, A.K. and H.G.; Software, A.K.; Validation, G.R.-B.O.; Formal analysis, H.G.; Writing—original draft, A.K.; Writing—review & editing, H.G. and G.R.-B.O.; Supervision, H.G. and G.R.-B.O. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Télécom Paris, l’Institut Polytechnique de Paris, France.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Datasets are available from the authors upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

ALS	Alternating Least Squares
AoD	Angle of Departure
AoA	Angle of Arrival
AWGN	Additive White Gaussian Noise
BA	Beam Alignment
BS	Base Station
BCE	Binary Cross Entropy
BCD	Block Coordinate Descent
BGD	Block Gradient Descent
BSGD	Block Stochastic Gradient Descent
CSI	Channel State Information
DFT	Discrete Fourier Transform
GD	Gradient Descent
LoS	Line of Sight
MF	Matrix Factorization
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NMF	Non-Negative Matrix Factorization
NLoS	Non Line of Sight
NMSE	Normalized Mean Squared Error
OFDM	Orthogonal Frequency Division Multiplexing
QoS	Quality of Service
ReLU	Rectified Linear Unit
RSE	Received Signal Energies
SNR	Signal-to-Noise Ratio
UE	User Equipment

Appendix A. Proof: BCD Convergence

We will show that the two (below) necessary conditions for convergence of BCD are satisfied:

- (i) The loss function is strongly convex, per block; i.e., we need to show that sub-problem S1 and S2 have a unique solution.
- (ii) The constraints of the MF prob $\theta_u \in \mathbb{R}^d$, $\psi_i \in \mathbb{R}^d$, are separable and individually convex.

Recall that sub-problem S1 is written as

$$(S1) : \theta_u^{(k+1)} = \operatorname{argmin}_{\theta_u \in \mathbb{R}^d} [-2\theta_u^T \mathbf{r}_u^{(k)} + \theta_u^T (\mathbf{Q}_u^{(k)} + \mu_u \mathbf{I}_D) \theta_u] = f_1(u), \quad \forall u,$$

Next, we will prove that the equivalent form in (S1), is a strongly convex function; i.e., it shows that $f_1(\theta_u)$ is strongly in θ_u . To that end, we derive the corresponding Hessian:

$$\nabla^2 f_1(\theta_u) := 2(\mathbf{Q}_u^{(k)} + \mu_u \mathbf{I}_D), \quad \forall u,$$

For this Hessian expression, $\mathbf{Q}_u^{(k)} \succeq \mathbf{0}$ is a Positive Semi Definite (PSD) matrix (by def), $\mu_u \mathbf{I} \succ \mathbf{0}$ is a Positive Definite (PD) matrix, and $(\mathbf{Q}_u^{(k)} + \mu_u \mathbf{I}_D) \succ \mathbf{0}$ is a PD matrix. Thus, the Hessian is a PD matrix $\nabla^2 f_1(\boldsymbol{\theta}_u) \succ \mathbf{0}$, and $f_1(\boldsymbol{\theta}_u)$ is strongly in $\boldsymbol{\theta}_u$, and the solution to the sub-problem (S1) is unique. Recall that the sub-problem (S2) is expressed as

$$(S2) : \boldsymbol{\psi}_i^{(k+1)} = \operatorname{argmin}_{\boldsymbol{\psi}_i \in \mathbb{R}^d} [-2\mathbf{t}_i^{(k+1)T} \boldsymbol{\psi}_i + \boldsymbol{\psi}_i^T (\mathbf{P}_u^{(k+1)} + \lambda_i \mathbf{I}) \boldsymbol{\psi}_i] = f_2(\boldsymbol{\psi}_i), \forall i,$$

Next, we will prove that the equivalent form is a strongly convex function; i.e., it shows that $f_2(\boldsymbol{\psi}_i)$ is strongly in $\boldsymbol{\psi}_i$. To that end, we derive the corresponding Hessian:

$$\nabla^2 f_2(\boldsymbol{\psi}_i) := 2(\mathbf{P}_i^{(k+1)} + \lambda_i^{(i)} \mathbf{I}_D), \forall i,$$

For this Hessian expression, $\mathbf{P}_i^{(k+1)} \succeq \mathbf{0}$ is a PSD matrix (by def), $\lambda_i^{(i)} \mathbf{I} \succ \mathbf{0}$ is a PD matrix, and $(\mathbf{P}_i^{(k+1)} + \lambda_i^{(i)} \mathbf{I}_D) \succ \mathbf{0}$ is a PD matrix. Thus, the Hessian is a PD matrix $\nabla^2 f_2(\boldsymbol{\psi}_i) \succ \mathbf{0}$, and $f_2(\boldsymbol{\psi}_i)$ is strongly convex in $\boldsymbol{\psi}_i$. Thus, the solution to the sub-problem (S2) is unique.

References

1. Wang, Y.; Wei, Z.; Feng, Z. Beam Training and Tracking in MmWave Communication: A Survey. *arXiv* **2022**, arXiv:2205.10169.
2. *IEEE Std 802.15.3c-2009*; IEEE Standard for Information Technology—Local and Metropolitan Area Networks—Specific Requirements—Part 15.3: Amendment 2: Millimeter-Wave-Based Alternative Physical Layer Extension. IEEE: Piscataway, NJ, USA, 2009.
3. *IEEE Std 802.11ad-2012*; IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band. IEEE: Piscataway, NJ, USA, 2012.
4. 3GPP. TS 38.211 V16.7.1 NR; Physical Channels and Modulation; ETSI Technical Specification 138 211 V16.10.0; Released: 07/2022. Available online: https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/16.10.00_60/ts_138211v161000p.pdf (accessed on 9 July 2024).
5. Noh, S.; Zoltowski, M.D.; Love, D.J. Multi-Resolution Codebook and Adaptive Beamforming Sequence Design for Millimeter Wave Beam Alignment. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 5689–5701. [\[CrossRef\]](#)
6. Kokshoorn, M.; Chen, H.; Wang, P.; Li, Y.; Vucetic, B. Millimeter Wave MIMO Channel Estimation Using Overlapped Beam Patterns and Rate Adaptation. *IEEE Trans. Signal Process.* **2016**, *65*, 601–616. [\[CrossRef\]](#)
7. Tsang, Y.M.; Poon, A.S.Y.; Addepalli, S. Coding the Beams: Improving Beamforming Training in mmWave Communication System. In Proceedings of the 2011 IEEE Global Telecommunications Conference—GLOBECOM 2011, Houston, TX, USA, 5–9 December 2011; pp. 1–6. [\[CrossRef\]](#)
8. Buzzi, S.; D’Andrea, C.; Subspace Tracking and Least Squares Approaches to Channel Estimation in Millimeter Wave Multiuser MIMO. *IEEE Trans. Commun.* **2019**, *67*, 6766–6780. [\[CrossRef\]](#)
9. Khordad, E.; Collings, I.B.; Hanly, S.V.; Caire, G. Compressive Sensing Based Beam Alignment Schemes for Time-Varying Millimeter-Wave Channels. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 1604–1617. [\[CrossRef\]](#)
10. Ghauch, H.; Skoglund, M.; Shokri-Ghadikolaei, H.; Fischione, C.; Sayed, A.H. Learning Kolmogorov Models for Binary Random Variables. In Proceedings of the International Conference on Machine Learning Workshop, Stockholm, Sweden, 10–15 July 2018.
11. Yetis, C.M.; Björnson, E.; Giselsson, P. Joint Analog Beam Selection and Digital Beamforming in Millimeter Wave Cell-Free Massive MIMO Systems. *arXiv* **2021** arXiv:2103.11199.
12. Dreifuerst, R.M.; Heath, R.W.; Yazdan, A. Massive MIMO Beam Management in Sub-6 GHz 5G NR. *arXiv* **2022**, arXiv:2204.06064.
13. Ma, K.; He, D.; Sun, H.; Wang, Z.; Chen, S. Deep Learning Assisted Calibrated Beam Training for Millimeter-Wave Communication Systems. *arXiv* **2021**, arXiv:2101.05206.
14. Nguyen, K.N.; Ali, A.; Mo, J.; Ng, B.L.; Va V.; Zhang, J.C. Beam Management with Orientation and RSRP using Deep Learning for Beyond 5G Systems. *arXiv* **2022**, arXiv:2202.02247.
15. Aldalbahi, A.; Shahabi, F.; Jasim, M. BRNN-LSTM for Initial Access in Millimeter Wave Communications. *Electronics* **2021**, *10*, 1505. [\[CrossRef\]](#)
16. Dehkordi, S.K.; Kobayashi, M.; Caire, G. Adaptive Beam Tracking based on Recurrent Neural Networks for mmWave Channels. *arXiv* **2021**. [\[CrossRef\]](#)
17. Hussain, M.; Michelusi, N. Learning and Adaptation for Millimeter-Wave Beam Tracking and Training: A Dual Timescale Variational Framework. *arXiv* **2021**. [\[CrossRef\]](#)
18. Dreifuerst, R.M.; Daulton, S.; Qian, Y.; Varkey, P.; Balandat, M.; Kasturia, S.; Tomar, A.; Yazdan, A.; Ponnampalam, V.; Heath, R.W. Optimizing Coverage and Capacity in Cellular Networks using Machine Learning. *arXiv* **2021**, arXiv:2010.13710.

19. Narengerile, N.; Thompson, J.; Patras, P.; Ratnarajah, T. Deep Reinforcement Learning-Based Beam Training for Spatially Consistent Millimeter Wave Channels. In Proceedings of the 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Helsinki, Finland, 13–16 September 2021; pp. 579–584. [[CrossRef](#)]
20. Wang, L.; Ai, B.; Niu, Y.; Gao, M.; Zhong, Z. Adaptive Beam Alignment Based on Deep Reinforcement Learning for High Speed Railways. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 19–22 June 2022; pp. 1–6. [[CrossRef](#)]
21. Ktari, A.; Ghauch, H.; Rekaya, G. Matrix Factorization for Blind Beam Alignment in Massive mmWave MIMO. In Proceedings of the 2022 IEEE Wireless Communications and Networking Conference (WCNC), Austin, TX, USA, 10–13 April 2022; pp. 2637–2642. [[CrossRef](#)]
22. Ktari, A.; Ghauch, H.; Rekaya, G. Cascaded binary classifiers for blind Beam Alignment in mmWave MIMO using one-bit quantization. In Proceedings of the International Conference on Communications (ICC), WS02 ICC23 Workshop, DDINS, Rome, Italy, 28 May–1 June 2023.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.