Machine Learning for beam Alignment in mmWave massive MIMO

Mohamed Aymen Ktari

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAT047

Thèse de doctorat

TELECOM Paris

IP PARIS

Machine Learning for beam Alignment in mmWave massive MIMO

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

Ecole Doctorale n° 626
Ecole Doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité: Réseaux, informations et communications

Thèse présentée et soutenue à Télécom Paris, Palaiseau, 19/12/2023, par

MOHAMED AYMEN KTARI

Composition du Jury :

Veronica Belmega Professeure, Université Gustave Eiffel, Paris, France	Présidente/Examinatrice
Matthieu Crussière Professeur, INSA Rennes, France	Rapporteur
Charly Poulliat Professeur, INP Toulouse, France	Rapporteur
Jean-Claude Belfiore Directeur, Advanced Wireless Technology Lab, Huawei Paris, France	Examinateur
Maxime Guillaud Directeur, INRIA Lyon, France	Examinateur
Ghaya Rekaya Professeure, Télécom Paris, France	Directrice de thèse

Machine Learning for beam Alignment in mmWave massive MIMO

Mohamed Aymen KTARI
TELECOM Paris, ED de l'Insitut Polytechnique de Paris

ACKNOWLEDGMENT

This doctoral dissertation was conducted at Telecom Paris, Institut Polytechnique de Paris, within the Comelec Department. I am deeply grateful to the laboratory team for their warm reception and unwavering support throughout my doctoral journey. I extend my heartfelt appreciation and sincere gratitude to my advisor and thesis director, Professor Ghaya Rekaya, for her exceptional guidance, tremendous patience, and invaluable expertise. In parallel, I would also like to gratefully acknowledge Assistant Professor Hadi Ghauch, my co-advisor, for his systematic approach, multidisciplinary assistance, and timely availability, all of which significantly contributed to the development of this work.

I am honored to express my sincere gratitude to the members of the jury, Pr. Veronica Belmega, Professor at Université Gustave Eiffel, Pr. Jean-Claude Belfiore, director of the Advanced Wireless Technology Lab at Huawei Paris, and Pr. Maxime Guillaud, Professor at INRIA Lyon, for their willingness to participate in my PhD defense and evaluate the quality of my doctoral research. Profound appreciation fills my heart as I extend my sincerest thanks to Pr. Matthieu Crussiere, Professor at INSA Rennes and Pr. Charly Poulliat, Professor at INP Toulouse for kindly accepting to examine my thesis and review the merits of my doctoral investigation.

I reserve special sentiments of appreciation for my family, whose unwavering support has been my constant source of strength, and for my friends, both within and beyond the academic sphere, whose encouragement has been instrumental in overcoming challenges.

My respectful gratitude goes to members of the mid-thesis jury, Pr. Frederic Lehmann and Pr. Matthieu Crussiere for kindly accepting to enrich and validate my mid-thesis defense.

Special words of love and gratitude are dedicated to the Chorale of Télécom Paris for three wonderful years of singing and rehearsing, where research and engineering meets music and arts throughout a passionate and fruitful PhD experience. I extend my appreciation to Pr. Michèle Wigger, assistant Pr. Robert Graczyk and Pr. Philippe Ciblat for their trust and guidance in my unforgettable teaching experience, regarding the digital communications and information theory TDs.

Finally, I extend my thanks to every individual who has played a role in enhancing my doctoral journey.

Résumé

Dans l'optique d'améliorer l'efficacité spectrale pour les réseaux 5G/6G, la technologie MIMO en ondes millimétriques a émergé, offrant des avancées significatives grâce à des techniques de précodage avancées. Malgré son potentiel, la complexité des environnements urbains réels et les caractéristiques uniques des fréquences mmWave posent des défis. La communication massive MIMO à des fréquences entre 30 GHz et 300 GHz nécessite un alignement précis des faisceaux, crucial pour établir des liens initiaux robustes. Les méthodes classiques dans les normes conventionnelles, telles que la WiGig, impliquent un sondage exhaustif des faisceaux, notre benchmark, entraînant une surcharge importante de signalisation de pilotes et l'impossibilité de le déployer dans des applications MIMO de grandes dimensions.

Notre recherche aborde ce problème en proposant un alignement partiel et aveugle des faisceaux, une approche qui intègre des techniques d'apprentissage automatique. En tirant parti des codebooks sous-échantillonnés et en utilisant des réseaux neuronaux et la factorisation matricielle, nous visons à réduire la surcharge des pilotes et à identifier avec précision les paires de faisceaux optimaux.

Dans la littérature, les méthodes de l'état de l'art sont divisées en deux familles: l'alignement classique des faisceaux et l'alignement basé sur l'apprentissage automatique. Les premières approches reposent sur l'alignement exhaustif des faisceaux, généralement via des codebooks hiérarchiques, la Compressed Sensing, le Beam Coding et de nombreux autres outils, visant à optimiser le processus d'alignement des faisceaux en utilisant tous les échantillons disponibles. Elles nécessitent généralement une estimation de canal basée sur l'échange des CSI tout en utilisant des architectures de précodage hybrides. Par contre, les approches basées sur l'apprentissage s'appuient sur moins d'échantillons d'entraînement avec des résultats prometteurs. Cependant, l'étude de la complexité du ML et de ses exigences matérielles illustre les défis liés à l'application d'outils d'IA dans les systèmes de communication sans fil.

Dans ce contexte, ce travail vise à investiguer la faisabilité de l'approche proposée d'alignement des faisceaux basée sur l'apprentissage automatique, en s'appuyant sur des architectures entièrement analogiques à faible complexité avec des chaînes RF limitées et des réseaux neuronaux peu profonds. Ainsi, nous avons commencé par un scénario basique, point-à-point, Uplink, à bande étroite et avons progressé, étape par étape, en formulant continuellement les nouveaux problèmes techniques et les contraintes rencontrées, dans le but de répondre mathématiquement et empiriquement à ces problématiques.

Nous avons d'abord considéré un scénario Uplink point-à-point, à bande étroite,

en champ libre, utilisant une chaîne RF pour l'utilisateur et une chaîne RF pour la station de base. La solution au problème d'alignement des faisceaux est basée sur l'application de la factorisation matricielle et de ses variantes pour accomplir la tâche d'Alignement Aveugle et Partiel des Faisceaux en utilisant des codebooks sous-échantillonnés. En plus des garanties théoriques, les résultats numériques servent de preuve convaincante, démontrant l'efficacité de notre approche hybride: "data-driven" et "model-based". Notamment, notre méthode atteint ses objectifs avec efficacité, en utilisant seulement 10% des faisceaux disponibles et en atteignant une solution entièrement aveugle au CSI. Cette réalisation représente une avancée significative pour relever le défi du grand surdébit de signalisation dans l'alignement des faisceaux et a conduit à la publication d'un article au sein de la conférence WCNC.

Notre deuxième découverte dans le cadre de ce doctorat a débuté par une extension de notre modèle système, augmentant la complexité de notre configuration expérimentale pour refléter des conditions réelles, notamment un modèle large bande en non-ligne de visée, plusieurs chaînes RF à la station de base et des codebooks DFT sous-échantillonnés. Par la suite, nous avons introduit l'architecture du perceptron multicouche, présentant ses équations d'entrée-sortie, énonçant le problème et formulant la solution. Ainsi, nous avons proposé une étude comparative, mettant en lumière l'interaction nuancée entre la complexité et la qualité des prédictions pour les deux méthodes. Nos résultats soulignent que seulement 10% de l'ensemble des paires de faisceaux sont suffisants pour aligner avec précision les faisceaux entre l'équipement utilisateur et la station de base, pour les deux méthodologies proposées, dans un scénario utilisateur unique point à point. De plus, nous avons exploré de manière perspicace les similitudes et les différences dans le comportement des modèles en variant la puissance émise. Ces résultats éclairent la viabilité pratique de nos méthodes, fournissant une base solide pour leur application dans des contextes réels et ouvrant la voie à une nouvelle ère d'alignement de faisceaux dans les futurs systèmes de communication. Ces découvertes ont permis la soumission du papier journal EURASIP.

Le troisième résultat de cette thèse de doctorat explore profondément la quantification, s'attaquant ainsi aux contraintes pratiques du déploiement des modèles d'apprentissage automatique. Nous avons commencé par établir l'architecture du système et la formulation des équations du modèle. Une illustration mathématique de ces contraintes a été entreprise, ouvrant la voie à la création du jeu de données quantifié, pour nos classifieurs. L'approche d'apprentissage a ensuite été élucidée, cartographiant la cascade des couches de régression logistique binaire et décrivant leurs équations d'entrée-sortie respectives à chaque étape.

Notamment, les révélations numériques ont mis en évidence une constatation cohérente et assertive: notre ratio optimal de surcharge (maintenu à 10% tout au long de cette thèse) peut s'harmoniser avec un schéma aggressif de quantification binaire. Cette adaptation, remarquablement, ne compromet pas la qualité prédictive tout en respectant des strictes prérequis de faible complexité, affirmant ainsi la praticabilité de notre approche proposée, comme illustré dans la publication du papier de la conférence ICC.

Enfin, nous avons investigué la gestion du scénario multi-utilisateur, en formulant

les problèmes sous-jacents et les équations du système. Nous avons ainsi présenté deux approches pour résoudre le problème: la complétion matricielle generalisée et la complétion tensorielle. Notre jeu de données, une collection de valeurs SINR pour chaque paire de faisceaux à travers tous les utilisateurs, est devenu la pierre angulaire sur laquelle notre procédure d'alignement a été élaborée, exploitant des codebooks sous-échantillonnés par DFT. À mesure que notre exploration approfondissait, les CNN peu profondes, MLP et AE ont non seulement atteint leurs objectifs, mais l'ont fait avec un faible ratio de surcharge de pilote, soulignant l'efficacité des deux approches proposées. De plus, le protocole expérimental a impliqué une comparaison axée sur la qualité de service, explorant les performances des modèles en fonction des échantillons d'entraînement disponibles. En reconnaissant les limitations rencontrées, nous avons tracé les voies pour des recherches futures, mettant en évidence des directions de recherches pour améliorer nos méthodologies et les éventuelles perspectives pour surmonter ces défis.

Contents

List of Figures

List of Tables

Glossary

AE: Auto-Encoder.

AI: Artificial Intelligence.

ALS: Alternating Least Squares.

AoD: Angle of Departure.

AoA: Angle of Arrival.

ADC: Analog to Digital Converter.

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.

CAE: Convolutional Auto Encoder.

CNN: Convolutional Neural Network.

DFT: Discrete Fourier Transform.

DAC: Digital to Analog Converter.

ERM: Empirical Risk Minimization.

FF: Feed Forward.

GD: Gradient Descent.

GPU: Graphics Processing Unit.

GPT: Generative Pre-trained Transformer.

GPS: Global Positioning System.

IEEE: Institute of Electrical and Electronics Engineers.

ISI: Inter Symbol Interference.

IP: Integer Programming.

LoS: Line of Sight.

LP: Linear Programming.

MDP: Markov Decision Process.

MF: Matrix Factorization.

ML: Machine Learning.

MRC: Maximum Ratio Combining.

MLP: Multi Layer Perceptron.

MIMO: Multiple Input Multiple Output.

MU-MIMO: Multi User Multiple Input Multiple Output.

MIP: Mixed Integer Programming.

MSE: Mean Squared Error.

MMSE: Minimum Mean Squared Error.

NMF: Non-Negative Matrix Factorization.

NLP: Non Linear Programming.

NLoS: Non Line of Sight.

NMSE: Normalized Mean Squared Error.

OFDM: Orthogonal Frequency Division Multiplexing.

PMF: Probability Mass Function.

PCA: Principle Component Analysis.

PDF: Probability Density Function.

QP: Quadratic Programming.

QoS: Quality of Service.

RMSE: Root Mean Squared Error.

ReLu: Rectified Linear Unit.

RNN: Recurrent Neural Network.

RSE: Received Signal Energies.

RF: Radio Frequency.

SGD: Stochastic Gradient Descent.

SotA: State-of-the-Art.

SVM: Support Vector Machine.

SNR: Signal to Noise Ratio.

SINR: Signal to Interference and Noise Ratio.

SU-MIMO: Single User Multiple Input Multiple Output.

Tanh: Hyperbolic Function.

TPU: Tensor Processing Unit.

t-SNE: t-distributed Stochastic Neighbor Embedding.

UE: User Equipment.

VAE: Variational Auto Encoder.

ZF: Zero Forcing.

Notations

The following notations are employed throughout the present PhD manuscript.

Notation	Description
a_i	i-th element of vector $\mathbf{a}$
a_i	i-th element of the random vector $\mathbf{a}$
$A_{i,j}$	Element at row i, column j of matrix $\mathbf{A}$
$A_{i,j}$	Element at row i, column j of the random matrix $\mathbf{A}$
$A_{i,:}$	Row i of matrix $\mathbf{A}$
$A_{:,j}$	Column j of matrix $\mathbf{A}$
$\mathbf{a}^T$	Transpose of vector $\mathbf{a}$
$\mathbf{A}^T$	Transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	Inverse of matrix $\mathbf{A}$
$\lvert\mathbf{A}\rvert$	Determinant of matrix $\mathbf{A}$
$\mathbf{A}^\dagger$	Transpose conjugate of $\mathbf{A}$, or Hermitian transpose
$\mathbf{A}^*$	Conjugate of $\mathbf{A}$
$diag(\mathbf{v})$	Diagonal matrix with diagonal elements from vector $\mathbf{v}$
$D(\mathbf{M})$	Vector (column) constructed from the diagonal elements of matrix $\mathbf{M}$
$A \times B$	Product of matrices $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{A} \odot \mathbf{B}$	Pointwise (Hadamard) product of matrices (or any n-dimensional tensor) $\mathbf{A}$ and B
$\lvert\lvert a\rvert\rvert_1$	1-norm of vector $\mathbf{a}$
$\lvert\lvert a\rvert\rvert_2$	2-norm, or Euclidean norm, of vector $\mathbf{a}$
$\lvert\lvert a\rvert\rvert_2^2$	Squared Euclidean norm of vector $\mathbf{a}$
$\mathbf{A} \succ 0$, $\mathbf{A} \succeq 0$	Positive Definite and Positive Semi-Definite matrices
$\mathbf{A} \prec 0$, $\mathbf{A} \preceq 0$	Negative Definite and Negative Semi-Definite matrices

Table 1: Algebra

Notation	Description
$\frac{df(x)}{dx} \in \mathbb{R}$ or $f'(x)$	Derivative of $f(x) : \mathbb{R} \to \mathbb{R}$ with respect to scalar x
$\frac{\partial f(x)}{\partial x} \in \mathbb{R}$	Partial derivative of scalar field $f(x) : \mathbb{R}^n \to \mathbb{R}$ with respect to scalar x
$\nabla x f(x) = \frac{\partial f(x)}{\partial x} \in \mathbb{R}^n$	Gradient of scalar field $f(x) : \mathbb{R}^n \to \mathbb{R}$ with respect to vector $\mathbf{x}$
$\nabla \mathbf{x} f\mathbf{x}(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \in \mathbb{R}^{n \times m}$	Matrix derivative of scalar field $f(\mathbf{X}) : \mathbb{R}^{n \times m} \to \mathbb{R}$ with respect to matrix $\mathbf{X}$
$\frac{\partial f(x)}{\partial x} \in \mathbb{R}^m$	Derivative of vector field $f(x) : \mathbb{R}^n \to \mathbb{R}^m$ with respect to scalar x
$J_x f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \in \mathbb{R}^{n \times m}$	Jacobian matrix of vector field $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ with respect to vector $\mathbf{x}$
$\frac{\partial F(X)}{\partial x} \in \mathbb{R}^{l \times p}$	Derivative of matrix function $F(X) : \mathbb{R}^{n \times m} \to \mathbb{R}^{l \times p}$ with respect to scalar x
$\nabla^2 x f(x) = \frac{\partial}{\partial x}(\nabla x f(x)) \in \mathbb{R}^{n \times n}$	Hessian of scalar field $f(\mathbf{x}\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ with respect to vector $\mathbf{x}$

Table 2: Calculus

Notation	Description
a	Scalar
$\mathbf{a}$	Vector (column)
$\mathbf{A}$	Matrix
$\mathbf{e}_k$	k-th standard basis vector, i.e., a vector with a 1 at index k and 0 otherwise
$\mathbf{I}_n$	Identity matrix of size $n \times n$. If n is not specified, it is implied by context
$\mathbf{0}$	All-zero vector (or any n-dimensional tensor) whose size is implied by context
$\mathbf{j}$	Imaginary unit
x^*	Complex conjugate of the complex number x

Table 3: Arrays

Notation	Description
a	Random scalar variable
$\mathbf{a}$	Random vector variable (column)
$\mathbf{A}$	Random matrix variable
$a \sim D$	Random variable a follows distribution D
$P(a \sim D \mid a = a)$	Probability that the random variable a takes the value a under distribution D
$p(a \sim D, a = a) = p(a)$	Probability Mass Function of (discrete) random variable a
$f(a \sim D, a = a) = f(a)$	Probability Density Function of (continuous) random variable a
$E(a \sim D)$	Expectation of the random variable a following distribution D
$Var(a \sim D)$	Variance of the random variable a following distribution D

Table 4: Probabilities

Notation	Description
$x(t)$	A continuous signal as a whole or indexed at time t
$x[k]$	A discrete signal as a whole or the k-th sample (sometimes denoted as x_k using vector indexing notations)
$\delta(x)$	Dirac function or Kronecker Delta function
$f(x;\theta)$	Parametric function of x with θ as a parameter
$\text{argmin}_\theta f(\theta)$	Argument of the minima of the function $f(\theta)$ with regard to parameters θ
$\text{argmax}_\theta f(\theta)$	Argument of the maxima of the function $f(\theta)$ with regard to parameters θ
$\lim_{x \to \infty} f(x)$	Limit of the function $f(x)$ as x approaches $+\infty$
$x * y$	Convolution of signal x by signal y (discrete or continuous)
$x \star y$	Cross-correlation of signal x by signal y (discrete or continuous)
R_{yy}	Auto-Correlation Matrix of signal y

Table 5: Functions

Notation	Description
S	A set
$\{1, 2, 3\}$	The elements of a set
S^n	The power set whose elements are the n-ary Cartesian product of S
S^c	Complement of a set (with regard to another)
$A \cup B$	Union of sets A and B
$A \cap B$	Intersection of sets A and B
$\text{Card}(S)$	The cardinal, i.e., size, of a set
$[a, b[$	An interval over an ordered set between a (included) and b (excluded) endpoints
$\mathbb{F}_p$	A finite field, i.e., Galois field, of p elements
$\mathbb{R}$	The set of real numbers
$\mathbb{C}$	The set of complex numbers

Table 6: Sets

Chapter 1

Introduction

"I propose to consider the question, Can machines think? This should begin with definitions of the meaning of the terms machine and think."

Alan Turing.

1.1 General context: 5G and towards 6G applications

In recent years, the exponential growth in data consumption and the emergence of bandwidth-intensive applications have driven the evolution of wireless communication systems for the fifth generation and towards the highly anticipated sixth generation. These next-generation networks aim to satisfy the increasing demand for higher data rates, ultra-low latency, and massive connectivity, ushering in a new era of communication for multiple applications including Enhanced Mobile Broadband, Internet of Things, mission-critical communications, augmented reality and autonomous vehicles. As predicted in [1], worldwide data traffic demand will grow to 5 Zettabytes per month, with personal data rates reaching 100 Gbps by 2030. To address these ambitious requirements, Multiple-Input Multiple-Output technology (MIMO) has is considered a fundamental pillar in enhancing system capacity and spectral efficiency. Specifically, Massive MIMO based technologies enhance spectral efficiency by allowing multiple users to be served simultaneously on the same frequency band offering better coverage, reduced interference, and enhanced overall system performance via diversity gain.

In particular, the utilization of millimeter-wave frequency bands has garnered substantial interest as a key resource for meeting the escalating data demands of future wireless networks. The mmWave bands, spanning from 30 to 300 GHz, offer

significantly wider bandwidths compared to traditional frequency bands, enabling unprecedented data transmission rates. By harnessing the vast spectrum available in the mmWave range, MIMO systems hold tremendous potential in transforming the wireless landscape and enabling a myriad of innovative applications.

Nevertheless, propagation at mmWave frequencies is severely impacted by high free-space path loss which results in a significant attenuation of transmitted signals in addition to remarkable penetration loss. These sensitive physical properties of mmWave suggest the use of large antenna arrays and the application of beamforming techniques in order to design highly directional beams. Hopefully, mmWave band is characterized by small wavelength which allows the implementation of massive antenna elements in small sized arrays in order to guarantee large beamforming gain. In this thesis, we investigate one fundamental problematic: what is the optimal beam steering direction between a transmitted and a receiver in mmWave Massive MIMO networks. This primordial technical challenge is denoted in the literature as the Beam Alignment (BA) problem:

- Technical problem: in order to guarantee a reliable initial link before data transmission, the beams at the transmitter and the receiver sides are continuously and constantly aligned and accurately steered. However, the alignment procedure is subject to a large pilot overhead, which scales with the resolution of the beamforming codebook and the number of antennas at both sides of the transmission, in crucial time-varying conditions and non-stochastic variations of the wireless network. We focus on the signaling overhead problem where traditional beam alignment techniques, such as exhaustive search based predefined codebooks are not applicable and are not well-suited for dynamic environments with high dimensional antenna systems.

- Proposed solution: to address this limitation, machine learning techniques have emerged as a promising solution, providing multiple tools that can adaptively learn beamforming patterns and optimize beam alignment based on channel conditions and environmental dynamics. Moreover, ML algorithms can rely on large-scale datasets, to extract complex beamforming features and strategies which aims to maximize signal quality, minimize destructive interference, and improve overall system performance.

1.2 Literature survey and technical overview

The beam alignment problem, due to its significance, has been extensively investigated and addressed in the existing literature. In conventional standards, Brute Force BA, is the de-facto approach for the Alignment process. Denoted as *Exhaustive* BA, it consists on sounding all beam-pairs at both sides of the transmission in order to Exhaustively select the beam couple with the highest Signal to Noise ratio. In the 60 GHz, this approach has been adopted in multiple mmWave $WLAN$ or $WPAN$ communication technologies, e.g., IEEE 802.15.3c [2], IEEE 802.11ad

[3] [4]. It is continuously and conventionally being applied in small MIMO configurations using small codebook sizes (e.g., codebooks of size 8×8 for LTE). Early approaches introduce two main families of BA methods: classical BA and ML-based BA.

- Classical BA: these techniques are all based on Exhaustive BA. They aim to require more and more structured Beam Alignment scheme using hierarchical multi-level codebooks [5] where training beamforming vectors with different beam widths on multiple stages are used. In addition, overlapped beam pattern [6] techniques, Beam coding [7] and Subspace estimation/decomposition based BA [8] are also well investigated in the literature. Besides, exploiting channel sparsity, Compressed sensing-based BA [9] estimates the angles of departure/arrival and the channel propagation path gains so that it constructs the beamforming vectors. Recently, a lot of researchers withdrew this approach due to non-linearity limitations. The limitations of classic approaches are the large signaling overhead ratio which states the impossibility of their deployment in massive MIMO systems. Some of these methods rely on strong assumptions regarding the temporal variations and generally require a precise prior knowledge of channel statistics, structure and sparsity.

- ML-based Beam Alignment: lately, Machine Learning tools for BA emerged and are increasingly and continuously illustrating promising results. For instance, statistical models such as Kolmogorov in [10] with sub-sampled codebooks introduced the concept of Partial BA where a small subset of all available beams is used for the Alignment process. Same core-theme explains the intensive use of multiple shallow neural networks where we can distinguish two major learning paradigms: first, Supervised Learning to resolve non-linear regression problems using Support Vector Machine for joint Analog beam selection in [11], convolutional neural networks based on a beam space observation in combination with RF environment characteristics in [12]. Similar neural architectures are used for calibrated beam training in [13] in addition to recurrent neural networks for beam tracking in [14][15][16] and auto-encoders for beam Management in [17]. Second, the Reinforcement Learning in [18][19][20], generally used to resolve the problems of Multi-Armed Bandit and Markov decision process. Recently, multiple semi-supervised and Unsupervised learning paradigms are increasingly investigated.

1.3 Research plan and manuscript organization

This PhD work focuses on addressing the challenge of large signaling overhead in Beam Alignment for mmWave massive MIMO systems using Machine Learning techniques. The goal is to propose signal processing approaches for Partial and Blind BA by involving sub-sampled codebooks that do not require explicit channel estimation. Our approach offers two main advantages: reducing the overhead by using a small

training set of beam pairs and suppressing the dependency on channel estimation using low-complexity ML methods for fully-analog system architecture.

The research methodology adopted in this thesis follows a progressive approach, starting from a basic and simple scenario and gradually incorporating constraints and complexity. At each stage, a specific problem is formulated and resolved, while considering the trade-off between accuracy and complexity from a Quality of Service perspective.

- The initial phase of the thesis involves an extensive literature survey and mathematical analysis of the problem. The massive MIMO system model begins with a simple architecture and progressively introduces additional constraints and complexities. We then formulate the non-convex optimization problem, known as Block Coordinate Descent, and explore its closed-form solution. Subsequently, the alignment process is framed as a matrix completion problem, where Matrix Factorization and its variants naturally fit the task at hand. For BA, the proposed approach involves using sounding Received Signal Energies (RSE), i.e., Received Signal Strength, to generate a dataset matrix, capturing RSE values for all beam-pairs between the User Equipment and Base Station. A small subset of this matrix is randomly selected as the training set for the ML models, while the remaining unsounded beams represent the test set. Matrix completion is performed by predicting the RSE values for the test set using ML techniques. Thus, we start the investigation proposing an Uplink, point-to-point, narrowband system model with fully-analog architecture using phase shifters. Supported by theoretical guarantees for the monotonic convergence of the Learning Cost function, the thesis empirically demonstrates the success of MF and Non-negative MF in the alignment task using only 10% fraction of the beam-pairs as the training set. The problem formulation shares similarities with collaborative filtering for recommendation systems [21], where MF aims to accurately complete sparse, massive, and low-rank dataset matrices. In order to investigate the impact of different optimizers and parameter constraints, six MF-based models/variants are implemented and showed promising results empirically.

- The system architecture in the second phase of the thesis is enhanced to include multiple RF chains at the Base Station, and the channel model follows the widely used Saleh-Valenzuela [22], from narrowband to wideband model with the utilization of Discrete Fourier Transform codebooks. This extension of the system model conducts the exploration of various Machine Learning tools for the same non-linear regression problem in mmWave massive MIMO systems. Additionally, the impact of varying the transmitted power (and so, the RSE regime) is investigated. ML contributions in this phase include the utilization of Multi-Layer Perceptron architecture and a QoS performance comparison with MF/NMF.

- In the third step of the thesis, a practical perspective is adopted, addressing the quantization of Received Signal Energy values before the alignment procedure.

This introduces a dual trade-off challenge: determining the minimum number of training samples required and identifying the optimal quantization scheme. Consequently, the dataset becomes discrete, and the problem transforms from non-linear regression to logistic regression. Empirical observations reveal that a binary quantization scheme using a cascaded structure of Binary Logistic Regression layers can achieve satisfactory performance with only 10% of the beams as the training set.

- The final research direction of the thesis focuses on the scalability of the proposed approach in a multi-user scenario. An uplink multi-user mmWave MIMO model is proposed based on Signal-to-Interference-plus-Noise Ratio values. The dataset now becomes a high-dimensional tensor, and two approaches are explored: processing each user separately using a shallow neural architecture or considering the entire tensor and feeding it to a denser and more complex neural network. This investigation aims to examine the trade-off between prediction quality and computational complexity in both approaches.

The manuscript follows a structured organization that aligns with the research plan. Chapter one provides an introduction to the PhD context and presents a brief survey of the state-of-the-art, emphasizing the large signaling overhead problem in massive MIMO systems and the need for ML techniques to overcome the limitations of conventional methods.

Chapter two introduces Machine Learning for Wireless Communications, encompassing a historical overview of AI and Computer Science, statistical learning principles, fundamentals of neural architectures, mathematical foundations of learning paradigms and optimization problems and an introduction to Matrix Factorization and its Non-negative MF variant.

Chapter three focuses on the Beam Alignment problem in mmWave MIMO communications. It begins with an overview of the mmWave band, highlighting its advantages, sensitive propagation properties, as well as limitations and challenges. The chapter then presents the Beam Management techniques, including Beamforming, precoding, combining, Beam Alignment, Beam Sweeping and Beam Tracking. The meeting between Machine Learning and Beam Alignment is introduced, emphasizing the resolution of non-linear and logistic regression problems throughout the PhD.

Chapter four constitutes the first contribution of the thesis, presenting Matrix Factorization for blind and partial Beam Alignment in massive mmWave MIMO systems. It includes the point-to-point narrowband system model, problem statement, benchmark, proposed solutions, algorithm, and numerical simulations for model training and test evaluation.

Chapter five represents the second contribution, introducing the utilization of Multi-Layer Perceptron for blind Beam Alignment in massive mmWave MIMO systems. The chapter encompasses an extended system architecture and a wideband model, problem formulation, proposed solutions, algorithm, numerical results, and a comparison with the performance of Matrix Factorization while considering variations in the transmitted power regime and signaling overhead ratios.

Chapter six illustrates the third contribution, incorporating practical constraints such as quantization and proposing a cascaded structure of Binary Logistic Regression layers for binary classification tasks. The chapter derives the total and aggregated signaling overhead and explores the accuracy/complexity trade-off through numerical simulations.

Chapter seven presents the fourth contribution, which introduces a multi-user system model based on SINR values and presents the utilization of Convolutional Neural Network for Multi-User Beam Alignment in mmWave MIMO systems. The chapter outlines the corresponding algorithm and experimental protocol to evaluate the performance of the proposed approach.

Finally, chapter eight concludes the thesis and provides perspectives for future research directions, highlighting the contributions made in this work and their implications for advancing the field of Beam Alignment in mmWave massive MIMO systems.

1.4 Publications

- Patent EP22305012, "Learning method for selecting beams in a Millimeter-Wave MIMO system", European Patent Office, patent filling in January 2022.
- A. Ktari, H. Ghauch and G. Rekaya, "Matrix Factorization for Blind Beam Alignment in Massive mmWave MIMO," 2022 IEEE Wireless Communications and Networking Conference (WCNC), pp. 2637-2642, Austin, Texas, April, 2022.
- A. Ktari, H. Ghauch and G. Rekaya, "Cascaded Binary Classifiers for Blind Beam Alignment in mmWave MIMO Using One-Bit Quantization," 2023 IEEE International Conference on Communications Workshops (ICC), pp. 80-85, Rome, Italy, June, 2023.
- A. Ktari, H. Ghauch and G. Rekaya, "Machine Learning techniques for blind Beam Alignment in mmWave Massive MIMO", submitted to EURASIP Journal on Wireless Communications and Networking, December, 2023.
- A. Ktari and G. Rekaya, "Neural Networks for multi user Beam Management in large dimensional mmWave MIMO", submitted to IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Madrid, July, 2024.

Chapter 2

Foundations of Machine Learning for Wireless Communications

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point."

Claude Shannon.

2.1 Introduction

Artificial Intelligence is a trending field of computer science focused on creating and designing systems capable of performing tasks that typically require human intelligence, such as problem-solving and learning from various and massive datasets. Machine Learning is a subset of AI that involves developing algorithms and models that enable computers to improve their performance on specific tasks through exposure to data and patterns, without being explicitly programmed and without any user intervention. On the other hand, Wireless Communications is a technology that enables the transmission of data and information over long distances without physical connections, using electromagnetic waves through the air. The relationship between AI and Wireless Communications lies in AI's potential to help the wireless industry by enhancing network management, optimizing resource allocation, combating crucial large signaling overheads and enabling intelligent decision-making, thereby advancing the capabilities and efficiency of wireless communication systems. In this preliminary chapter, we embark a historical journey to explore the evolution of Artificial Intelligence. Subsequently, we delve into the foundational pillars of AI, encompassing essential elements like Statistical Learning, Probability theory, Neural Networks, the Empirical Risk Minimization principle, various Learning paradigms, Optimization problems, and Matrix Factorization for completing low-rank matrices.

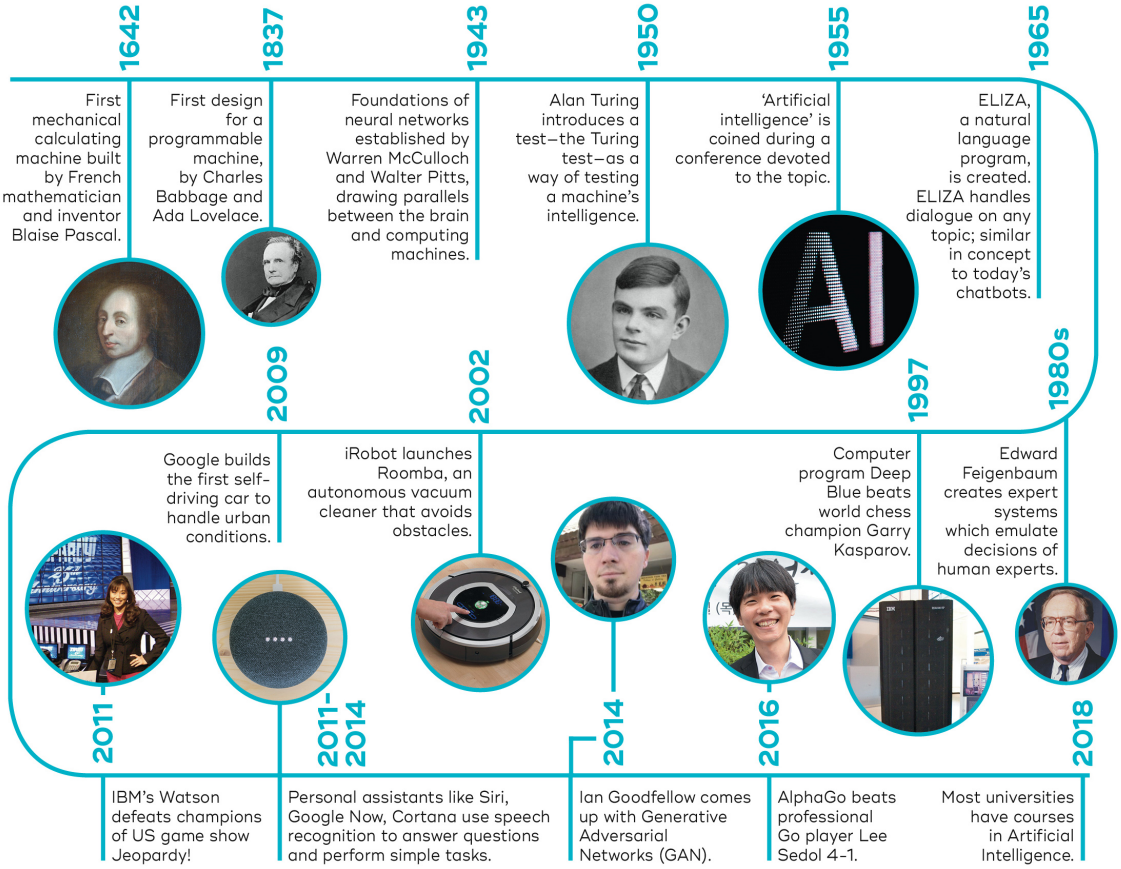


Figure 2.1: AI timeline [29]

2.2 A brief history of Artificial Intelligence and Computer Science

The fields of Artificial Intelligence and Computer Science have experienced significant progress throughout their extensive timeline, profoundly influencing various aspects of our society. In the course of history, notable advancements can be traced back to ancient times, such as the analog computer known as the Antikythera mechanism from the second century BC [24], as well as the development of Euclid's algorithm for calculating the greatest common divisor of integers [25]. The desire to automate machines has been a longstanding aspiration of humankind, and visionaries like Charles Babbage (1791 - 1871) [26], Ada Lovelace (1815 - 1852) [27], and Alan Turing (1912 - 1954) [28] played pivotal roles in advancing this dream. Their contributions marked a significant shift from single-purpose computing devices to the advent of general-purpose computers. The modern era of AI can be traced back to the Dartmouth Conference in 1956, where influential researchers like John McCarthy and Marvin Minsky coined the term "Artificial Intelligence" and laid the foundation for future advancements.

- Since then, AI has experienced periods of significant progress and transfor-

mative breakthroughs, alongside phases of reduced funding and diminished interest, commonly referred to as AI winters. However, in the 21st century, AI has experienced a resurgence, fueled by exponential growth in computational power, the availability of vast amounts of data, and groundbreaking algorithmic innovations.

- The timeline of AI breakthroughs showcases notable milestones. In the 1990s, the field witnessed the rise of machine learning, particularly with the introduction of neural networks and the development of efficient learning algorithms. This period saw the successful application of AI techniques in various domains, including natural language processing, computer vision, and speech recognition.

- In more recent years, the advent of big data and the proliferation of powerful computing systems have enabled the training of deep neural networks, giving rise to the era of deep learning. This advancement has revolutionized AI by enabling breakthroughs in areas such as image and video recognition, autonomous vehicles, and natural language understanding

- Simultaneously, Computer Science as a discipline has evolved alongside these AI developments. It encompasses various sub-fields, including algorithms, data structures, programming languages, software engineering, and computer networks. The evolution of Computer Science has been tightly intertwined with technological advancements, such as the development of powerful computers, the establishment of the Internet, and the proliferation of digital devices.

- These advancements in AI and Computer Science have reshaped the way we process information, communicate, and interact with the world. They have paved the way for transformative applications and technologies, ranging from intelligent virtual assistants and autonomous systems to data-driven decision-making and smart infrastructure.

Computers have long surpassed humans in solving complex numerical problems. However, more recently, these algorithms have started to outperform humans in more intricate and generalized tasks. For example, while a human annotator achieves a top-5 accuracy of only 5.1% on the *ImageNet* [30] image classification challenge, the best model achieved a remarkable top-5 accuracy of 99.02%. This remarkable progress has been made possible by the widespread availability of GPU-accelerated computation and vast datasets. *Narrow AI* models, also known as state-of-the-art models due to their specialization in specific tasks, are paving the way towards the ultimate goal of achieving *general AI*. Remarkable advancements in this multidisciplinary field are evident in groundbreaking models such as *DeepMind Alpha-Go* [31] and *OpenAI GPT-3* [32], two decades after Alan Turing's historical seminal paper "Can Machine Think?" [33]. These advancements serve as a testament to the remarkable strides made in this rapidly evolving field.

In terms of Telecommunication applications, the integration of AI techniques with wireless communication systems has opened up new possibilities for intelligent

and efficient wireless networks. By leveraging AI algorithms, such as statistical learning and deep learning, wireless communication systems can adapt dynamically to changing environments, optimize resource allocation, overcome the large signaling overhead problems, ensure robustness and improve overall system performance. This integration of AI and wireless communications holds immense potential for future 6G applications, enabling higher data rates, lower latency, and more intelligent network management.

2.3 Statistical Learning and foundations of probability

Statistical learning, a prominent field in both statistics and machine learning, focuses on the development of computational tools and techniques for making informed decisions and predictions from data. It provides a framework for extracting meaningful insights, identifying patterns, and building models that capture the underlying relationships within complex datasets. At its core, statistical learning leverages statistical principles and methodologies to uncover valuable information and make reliable inferences in the presence of uncertainty.

In statistical learning, the primary objective is to understand and model the relationship between the input variables, often referred to as features or predictors, and the output variables, known as responses or targets. This involves analyzing the patterns, trends, and dependencies present in the data to develop models that can be used for prediction, classification, and inference. Statistical learning approaches encompass a range of methodologies, including regression analysis, classification algorithms, re-sampling techniques, and dimensionality reduction methods, among others. These techniques are grounded in statistical theory and rely on probability distributions, hypothesis testing, and estimation principles to make data-driven decisions.

Machine learning, on the other hand, is a broader field that encompasses statistical learning as one of its key components. While statistical learning focuses on modeling and inference, machine learning extends beyond this to include the development of algorithms and computational systems that can automatically learn from data and improve their performance through experience. Machine learning algorithms are designed to identify patterns, extract knowledge, and make predictions without explicit programming instructions. They leverage computational power and advanced mathematical techniques to automatically learn from data, adapt to changing environments, and make accurate predictions or decisions.

Probability background:
Probabilistic reasoning is based on dealing with variables that have the capability of assuming different values at random. The technical general objective is modelling natural phenomenons relying on measuring Uncertainty. These variables are referred to as random variables and illustrate the core of Probabilities. A random variable, on its own, may not be particularly useful as it merely enumerates possible outcomes. However, when linked with a probability distribution, it becomes possible to

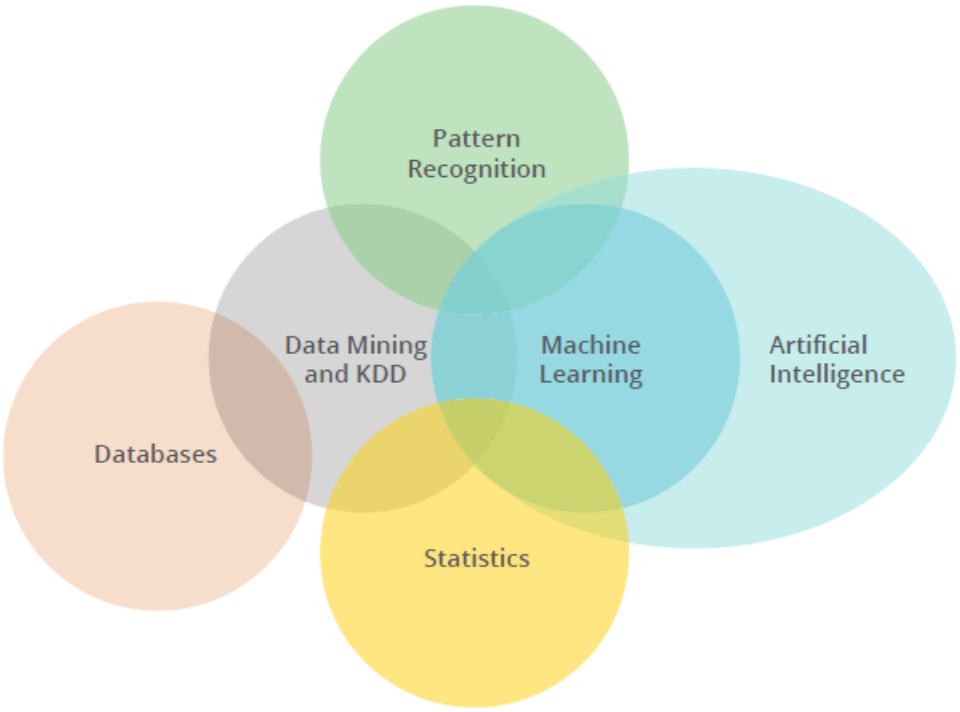


Figure 2.2: Statistical learning intersections with AI and ML disciplines

describe the likelihood of these outcomes occurring, thereby providing insights not only into what is possible but, more crucially, what is likely. A random variable can be defined as a variable that takes on diverse, either discrete or continuous, values randomly within a sample space Γ. In this manuscript, we denote a scalar random variable as x, while random vectors are represented by $\mathbf{x}$ and random matrices are represented by $\mathbf{X}$ where the probability distribution of a random variable characterizes the probability of each of its possible outcomes. Within this document, the notation $x \sim D$ is used to indicate that the random variable x follows the probability distribution D.

In the case of a discrete variable, such as when describing the outcome of a coin toss, the distribution of the random variable is typically expressed using a Probability Mass Function, denoted as $p_{x\sim D}(\mathrm{x} = x)$ or simply $p_x(x)$. This function quantifies the probability of the random variable x assuming the value x under the distribution D. A probability of 0 indicates that it is impossible signifies that an outcome is impossible while a probability of 1 illustrates that the outcome is certain. Therefore, a PMF must adhere to the following properties:

1. $0 \leq p_x(x) \leq 1, \forall x \in \Gamma$, the probability of all outcomes lies between 0 and 1.
2. $\sum_{x\in\Gamma} p_x(x) = 1$, the sum of the probabilities of all outcomes is equal to 1.

When dealing with continuous random variables, the sample space is linked to one or multiple continuous intervals over the infinite set of real numbers $\mathbb{R}$. The probability distribution is then characterized using the continuous Probability Density

Function, denoted as $f_{x\sim D}(x = x)$ or simply $f_x(x)$. Since the function is continuous, the PDF $f_x(x)$ does not directly provide the absolute likelihood of the random variable x being equal to x, as this would tend towards 0 due to infinitely many possible outcomes. Instead, the PDF represents the relative likelihood or the probability of the random variable x taking on a value in the infinitesimal neighborhood dx of x, expressed as $f_x(x)dx$. In a parallel way to PMF, the PDF is defined through two major properties:

1. $\int_{x\in\Gamma} f_x(x)dx = 1$, where we switch from discrete summation to an integral. Similarly to the PMF, the sum of all outcomes must sum to 1.

2. $0 \leq f_x(x), \forall x \in \Gamma$, PDF is by definition always positive

Additionally, for any specific interval $[a, b]$ within the sample space Γ, the probability of the random variable x falling within that interval can be determined by integrating the PDF over that interval. Mathematically, this can be expressed as $P_x(a \leq x \leq b) = \int_a^b f_x(x)dx$

These characteristics guarantee that the PDF provides a valid and meaningful representation of the probabilities associated with the continuous random variable x, allowing us to compute the likelihood of the variable falling within different intervals and making probabilistic predictions.

Another fundamental definition in the domain of probabilities is the *marginal distribution*, calculated using the sum rule in order to handle problems related to a joint distribution over a sample-subset of the total variables. For instance, for a set of n discrete random variables $\{x_1, ..., x_n\}$, the marginal distribution over the first $k < n$ variables is expressed as:

$$p_{x_1,...,x_k}(x_1, ..., x_k) = \sum_{x_{k+1}\in\Gamma_{k+1}} ... \sum_{x_n\in\Gamma_n} p_{x_1,...,x_n}(x_1, ..., x_n) \tag{2.1}$$

In case we handle continuous variables:

$$f_{x_1,...,x_k}(x_1, ..., x_k) = \int_{x_{k+1}\in\Gamma_{k+1}} ... \int_{x_n\in\Gamma_n} f_{x_1,...,x_n}(x_1, ..., x_n)dx_n...dx_{k+1}, \tag{2.2}$$

On the other hand, when an event A is conditioned by another event B, we introduce the notion of *conditional* probability where the probability of the realization of event A is related to the assumption that event B is realised. The conditional probability of event A knowing B is formulated as:

$$P\{A|B\} = \frac{P\{A, B\}}{P\{B\}} \text{ for } P\{B\} > 0 \tag{2.3}$$

The *Bayes* rule is one of the fundamentals in the literature and is derived from the conditional probability definitions. It computes the probability of an Hypothesis H knowing an observation of Evidence E using the following equation:

$$P\{H|E\} = P\{E|H\}\frac{P\{H\}}{P\{E\}} \tag{2.4}$$

where $P\{H\}$ is the *prior* i.e. the probability that hypothesis $\{H\}$ is true before any observed evidence while $P\{E\}$ is denoted as the *marginal likelihood* i.e. the probability of seeing the evidence $\{E\}$. $P\{E|H\}$ is known as the *likelihood* or the probability to observe the evidence $\{E\}$ given the hypothesis $\{H\}$. Ultimately, $P\{H|E\}$ is the *posterior* i.e. the probability that hypothesis $\{H\}$ is true given the evidence $\{E\}$.

Besides, we derive the product rule of probability where a multivariate joint distribution is factored into the chain product of the corresponding conditional probability:

$$P\{X_1, ..., X_n\} = P\{X_1\} \prod_{i=2}^{n} P\{X_i|X_1, ..., X_{i-1}\} \tag{2.5}$$

Moreover, the *expectation*, often denoted in the literature as the first moment, is defined as the weighted average of a (large) number of samplings according to their probability of occurrence and is expressed as follows for a discrete variable

$$E_{x\sim D}\{x\} = \sum_i x_i p_{x\sim D}(x = x_i) \tag{2.6}$$

When we handle continuous variables, the expectation is formulated similarly as follows:

$$E_{x\sim D}\{x\} = \int x f_{x\sim D}(x = x)dx \tag{2.7}$$

On the other hand, we mathematically introduce the relationship between probabilistic modeling and inference which serves as mathematical background for various ML models. We then denote $\mathbf{d}$, the generated sequence of samples regarding an unknown stochastic process and θ the parameters of the probabilistic model. Therefore, we introduce the *likelihood function*, denoted as $\mathcal{L}(\theta, \mathbf{d})$, representing the probability that the model (with parameters θ) outputs a sequence identical to the observed sequence $\mathbf{d}$:

$$\mathcal{L}(\theta, \mathbf{d}) = P_{d\sim D}\{\mathbf{d} = d; \theta\} \tag{2.8}$$

where $P_{d\sim D}\{\mathbf{d} = d; \theta\}$ represents the sampled stochastic sequence $\mathbf{d}$ from distribution D and have the same values as the observed realizations of the unknown data sampling distribution. When samples are independent, we define the *log-likelihood* function as the following product of probabilities:

$$\log \mathcal{L}(\theta, \mathbf{d}) = \log \prod_{i=1}^{n} P_{d_i\sim D}\{d_i = d_i; \theta\} = \sum_{i=1}^{n} \log P_{d_i\sim D}\{d_i = d_i; \theta\} \tag{2.9}$$

Note that the logarithm is strictly monotonic function which means it has no effect on minimization or maximization procedures and recall that the core of ML is the minimization of the Cost, i.e. Loss, function defined in the next sections of this chapter.

The intersection between statistical learning (based on probability reasoning) and machine learning lies in their shared goal of utilizing data to extract valuable information and make predictions or decisions. Both fields employ mathematical and statistical techniques to analyze datasets and build models. However, machine learning tends to emphasize computational efficiency, scalability, and the ability to handle large and complex datasets. It encompasses a broader range of algorithms, including both statistical and non-statistical methods, such as deep learning, reinforcement learning, and ensemble methods. Machine learning also places greater emphasis on algorithmic design, optimization, and the deployment of models in real-world applications.

While statistical learning and machine learning share common goals and principles, their differences primarily lie in their historical roots, methodologies, and emphases. Statistical learning often provides a more interpretable and transparent framework, making it well-suited for domains where model interpretability and inference are crucial. Machine learning, with its focus on automation, scalability, and complex modeling techniques, excels in scenarios where predictions or decisions need to be made rapidly and accurately, even in the absence of a deep understanding of underlying statistical relationships. Both fields contribute to the advancement of data-driven decision-making and play vital roles in various scientific, industrial, and societal domains. Recall that in this PhD manuscript, ML is used to solve Beam Alignment problem through a ML model-based and data-driven approach.

2.4 Neural networks

Neural architectures serve as the foundation of modern artificial intelligence systems, mimicking the structure and functionality of the human brain.

At the core of neural architectures are artificial neurons, also known as perceptrons, which receive inputs, apply mathematical operations, and produce output signals. These artificial neurons are organized into layers, with each layer performing specific computations. The input layer receives raw data, which is then processed through multiple hidden layers, each consisting of interconnected neurons. The final layer, known as the output layer, generates the desired output or prediction. The connections between neurons are defined by weights, representing the strength of the connections. During training, these weights are adjusted iteratively using optimization algorithms, such as gradient descent, to minimize the error between predicted and actual outputs.

In this subsection, we delve into the foundations of neural networks by introducing the formal neuron, which serves as the building block for more complex multi-layer architectures. By understanding the formal neuron, we lay the groundwork for exploring deep learning concepts and techniques including gradient descent, backpropagation and activation functions.

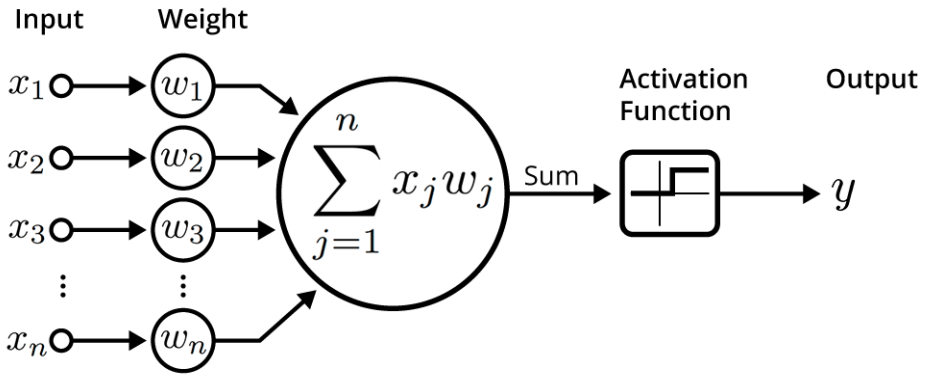


Figure 2.3: Formal neuron diagram representation

2.4.1 From formal neuron to increasing in depth

From the formal neuron, the field of neural architectures has witnessed significant advancements, particularly in increasing the depth and complexity of these networks. While the initial models consisted of a single layer of neurons, known as the perceptron, researchers soon recognized the limitations of such shallow architectures in capturing complex relationships within the data.

A formal neuron, also known as a perceptron, is a fundamental building block of neural architectures. It takes a set of input values, each multiplied by a corresponding weight, and applies a mathematical operation, typically a weighted sum, to produce an output. The output is then passed through an activation function, which introduces non-linearity into the neuron's response. The activation function helps determine whether the neuron should "fire" or be inactive based on the input. This firing decision is often represented as a binary output, where the neuron outputs 1 if the activation threshold is exceeded, and 0 otherwise.

To perform the calculations inside a neuron, let's consider an example with three input values, x_1, x_2, and x_3, and their corresponding weights, w_1, w_2, and w_3. As shown in figure (2.3), the weighted sum is computed as the sum of each input multiplied by its weight:

$$z = w_1 x_1 + w_2 x_2 + w_3 x_3 \tag{2.10}$$

The weighted sum, z, is then passed through the activation function, which maps the output to a desired range. Common activation functions, given in table (2.1), include the step function, sigmoid function, and rectified linear unit function. The choice of activation function depends on the specific problem and the desired properties of the neuron's response.

While perceptrons and shallow neural architectures could be powerful tools in some specific tasks, they have certain limitations. One major limitation is their inability to capture complex relationships or patterns in the data. This is because perceptrons are linear classifiers and can only separate data points with a straight line or plane. To overcome this limitation and enable more sophisticated learning, increasing the depth of neural architectures is necessary. By adding more hidden

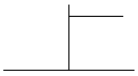
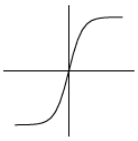
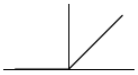
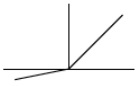
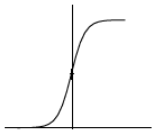
Name	Symbol	Definition	Derivative	Figure
Step	$H(x)$	$\begin{cases} 0 \text{ if } x < 0 \\ 1 \text{ if } x \geq 0 \end{cases}$	$\delta(x)$	
Sigmoid	$\sigma(x)$	$\frac{1}{1+e^{-x}}$	$\sigma(x)(1-\sigma(x))^2$	
Hyperbolic Tangent	$\tanh(x)$	$\frac{e^{x}-e^{-x}}{e^{z}+e^{-z}}$	$1-\tanh(x)^2$	
ReLU	$\mathrm{R}(x)$	$\max(0,x)$	$\begin{cases} 0 \text{ if } x < 0 \\ 1 \text{ if } x \geq 0 \end{cases}$	
Leaky ReLU	$\acute{\mathrm{R}}(x)$	$\max(\alpha x, x)$ $\forall \alpha \in [0,1]$	$\begin{cases} \alpha \text{ if } x < 0 \\ 1 \text{ if } x \geq 0 \end{cases}$	
Softmax	$\sigma_i(\boldsymbol{x})$	$\frac{e^{x_i}}{\sum_j e^{x_j}}$	$\frac{\partial \sigma_i(x)}{\partial x_k} = \begin{cases} \frac{e^{x_i}\sum_{j\neq i} e^{x_j}}{\left(\sum_j e^{x_j}\right)^2} \text{ if } k = i \\ -\frac{e^{x_i}e^{x_k}}{\left(\sum_j e^{x_j}\right)^2} \text{ if } k \neq i \end{cases}$	

Table 2.1: Frequently used activation functions: definitions, derivatives and plots

layers and neurons, neural networks can learn hierarchical representations of data, capturing intricate relationships and patterns that would be challenging for a single perceptron. This increased depth allows for more expressive models and enables the network to learn complex features and make accurate predictions.

2.4.2 Activation functions and neural architectures

Activation functions play a crucial role in neural architectures by introducing non-linearity into the output of individual neurons. They determine whether a neuron should be activated or remain inactive based on the input it receives. Various activation functions are used in neural networks, each with its own characteristics and suitability for different types of problems. Table (2.1) states a mathematical comparison of widely used activation functions:

- One commonly used activation function is the step function, which produces a binary output. It maps input values below a certain threshold to 0 and input values above the threshold to 1. The step function is useful for binary classification problems where the neuron's output needs to be a discrete decision.

- Another popular activation function is the sigmoid function, also known as

the logistic function. It maps the input to a value between 0 and 1, representing the probability of the neuron being activated. The sigmoid function is advantageous because it provides a smooth transition from 0 to 1, allowing for more gradual changes in neuron activations. It is commonly used in problems involving probability estimation and binary classification tasks.

- The rectified linear unit function is another widely used activation function, especially in deep learning architectures. It sets the output to 0 for negative input values and retains positive input values as they are. The ReLU function introduces sparsity and non-linearity into the network, enabling better gradient flow during backpropagation and faster convergence. It has been found to be particularly effective in handling large-scale datasets and deep neural networks.

- Other activation functions include the hyperbolic tangent function, which maps the input to a value between -1 and 1, and the softmax function, commonly used in multi-class classification problems to produce a probability distribution over multiple classes.

The choice of activation function depends on the specific problem at hand, the desired properties of the neuron's response, and the characteristics of the dataset. Selecting an appropriate activation function is essential for ensuring that the neural network can effectively learn and represent complex relationships in the data.

On the other hand, there are various types of neural architectures commonly used in artificial intelligence and machine learning. These architectures are designed to address different types of problems and have varying levels of complexity. Some of the commonly used neural architectures include feed forward neural networks, recurrent neural networks, convolutional neural networks, auto-encoders and generative adversarial networks. Each architecture has its own unique characteristics and is suited for specific tasks. The choice of architecture depends on the nature of the data and the problem at hand. Figure (2.4) plots an overview of several extensively used neural networks.

- Feed forward neural network, also known as a multi layer perceptron, consists of an input layer, one or more hidden layers, and an output layer. It processes information in a forward direction, with no loops or feedback connections, making it suitable for tasks such as classification and regression.

  Mathematically, we denote $h^{(j)}$ the output from each layer (j), $\sigma^{(j)}$ the activation function at layer (j). $\mathbf{W^{(j)}}$ is the Weights matrix and $\mathbf{x^{(j)}}$ is the bias vector of the j-th layer of the feed forward architecture, containing l dense layers. The input-output equation is expressed as:

$$h^{(1)} = \sigma^{(1)}(\mathbf{W^{(1)}}x) + \mathbf{b^{(1)}} \tag{2.11}$$

$$h^{(j)} = \sigma^{(j)}(\mathbf{W^{(j)}}h^{(j-1)}) + \mathbf{b^{(j)}}, \forall j \in \{2, ..., l-1\} \tag{2.12}$$

$$y = \sigma^{(l)}(\mathbf{W^{(l)}}h^{(l-1)}) + \mathbf{b^{(l)}} \tag{2.13}$$

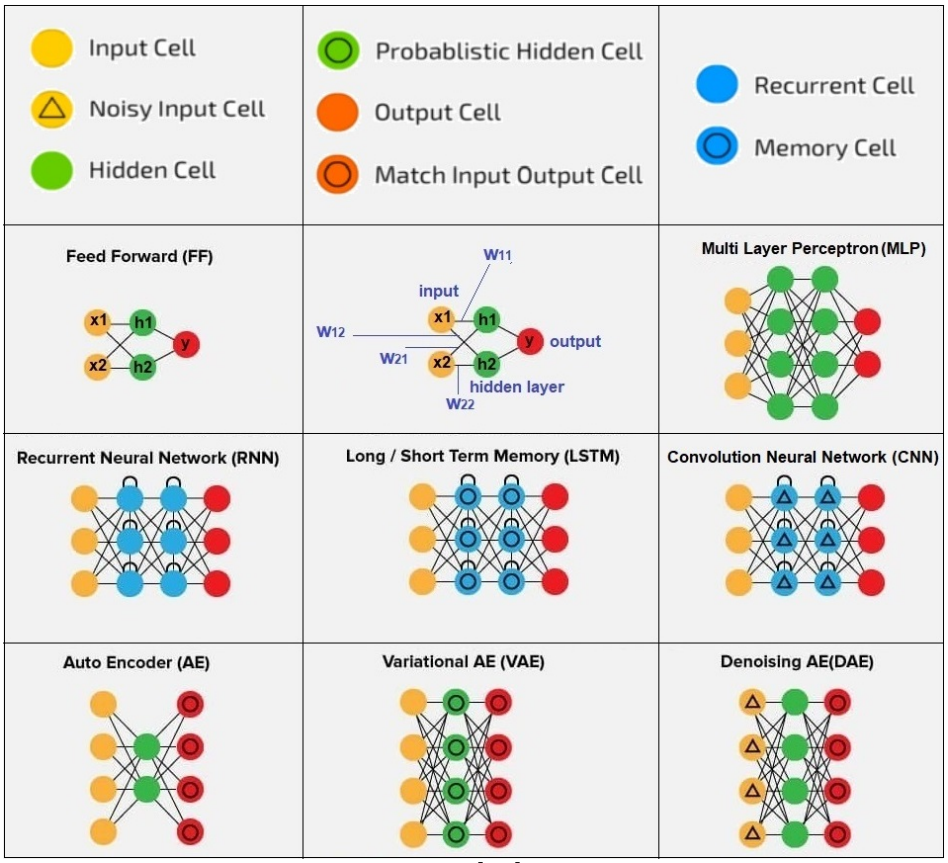


Figure 2.4: Neural architectures overview [34]: the distinction between neural architectures lies in the specific types of cell functions that constitute individual layers and the manner in which these layers are arranged and interconnected.

- Auto encoder is a type of neural network that aims to reconstruct its own input data. It consists of an encoder network that compresses the input data into a lower-dimensional representation, and a decoder network that reconstructs the original input from the compressed representation. Auto encoders are often used for dimensionality reduction and unsupervised learning tasks. Similarly, variational auto encoder is a generative model that learns to approximate the underlying probability distribution of the input data and enables the generation of new samples from the learned distribution.

- Recurrent neural network is designed to process sequential data by utilizing recurrent connections that allow information to be carried forward in time. It is well-suited for tasks such as natural language processing, speech recognition, and time series analysis. One commonly used RNN is the long short term memory which addresses the vanishing gradient problem and allows for better capturing long-term dependencies in sequential data.

- Convolutional neural network is primarily used for analyzing visual data, such as images. It consists of convolutional layers that extract local features from the input data, pooling layers that down-sample the features, and fully connected layers for classification or regression. CNNs have achieved remarkable success in tasks such as image classification, object detection, and image generation.

The convolution is a mathematical operation that involves combining two functions to produce a third function that represents the interaction between them. In the context of signal processing, convolution is used to process signals by applying a filter or kernel to it. This filter is usually a small window of values that slides over the input data, and at each position, the element-wise multiplication of the filter's values with the corresponding values in the input data is summed up to produce the output value at that position in order to capture local patterns and features in the input data. For instance, given two continuous signals x and h, the convolution of x by h is the integral of the two signals product after one is reversed and shifted, generalized over all shift values:

$$y(\tau) = (x * h)(\tau) = \int_{t=-\infty}^{+\infty} x(t)h(\tau - t)dt \tag{2.14}$$

If the x and h signals are discrete, the convolution product is similarly defined as:

$$y[n] = (x * h)[n] = \sum_{m=-\infty}^{+\infty} x[m]h[n - m] \tag{2.15}$$

In the literature, the *cross-correlation* operator, like the convolution operator, asses the degree of similarity of two continuous signals:

$$R_{xy}(\tau) = (x * y)(\tau) = \int_{t=-\infty}^{+\infty} x(t) * y(t - \tau)dt \tag{2.16}$$

Similarly when x and y are discrete signals:

$$R_{xy}[n] = (x * y)[n] = \sum_{m=-\infty}^{+\infty} x[m]y[m-n] \tag{2.17}$$

We distinguish 1D convolutional layer where the kernel, input, output and bias are vectors from 2D convolutional layer where these mathematical entities are matrices. For instance, we consider a kernel of size 2 (two coefficients $k1$, $k2$ in the kernel vector). The computations with 1D-convolutional dense layer are represented in matrix form as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_m \end{bmatrix} = \sigma \left(\begin{bmatrix} k_1 & k_2 & 0 & ... & ... & 0 \\ 0 & k_1 & k_2 & ... & ... & ... \\ : & : & : & : & : & : \\ ... & ... & ... & k_1 & k_2 & 0 \\ 0 & ... & ... & 0 & k_1 & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ : \\ : \\ : \\ x_n \end{bmatrix} + \begin{bmatrix} b \\ b \\ : \\ : \\ : \\ b \end{bmatrix} \right)$$

The 1D-convolution input-output equation is expressed as:

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \equiv \mathbf{y} = \sigma(\mathbf{k} * \mathbf{x} + \mathbf{b}) \tag{2.18}$$

On the other hand, the core concept of 2D-convolution is simplified and resumed in figure (2.5). In this example, the filter is a 2D matrix of size 3×3 and is applied on a 2D input matrix of size 6×6. We consider no padding, no bias and no activations for the sake of simplicity while fixing the stride as equal to one, the resulted output matrix is of size 4×4.

2.4.3 Empirical Risk Minimisation principle and Loss functions

Unlike classical optimization algorithms, Machine Learning methods rely on minimizing a surrogate error function substituting the actual error function in the optimization problem. The surrogate loss function, often denoted as surrogate objective function, is useful when the original loss function is complex or computationally expensive as they can simplify the optimization process without sacrificing the quality of the final solution. Common examples of surrogate error functions include hinge loss for support vector machines, soft max cross-entropy loss for neural networks in classification tasks and squared loss for regression problems. These functions are chosen based on their mathematical properties and suitability for the optimization algorithms being used.

Conventionally, we define $\mathcal{S} = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=1}^{s}$ as a standard training set of size s containing all sample pairs characterized by $\mathbf{x_i}$ as the features vector and $\mathbf{y_i}$ as the corresponding labels. Therefore, we define a mapping function $f(\mathbf{x_i}, \mathbf{\Theta})$ from inputs $\mathbf{x_i}$ to known outputs, labels, $\mathbf{y_i}$. Obviously, $\mathbf{\Theta}$ represents the parameters

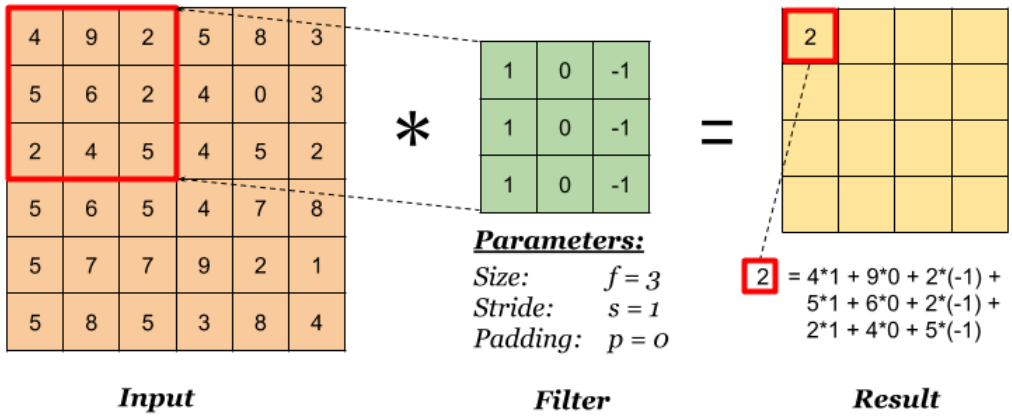


Figure 2.5: 2D convolution layer simplified diagram representation [35]: Size represents the dimensions of the convolutional filter, Padding consists in adding extra border pixels to the input image to control the spatial dimensions of the output feature map after convolution and Stride is the step size at which the convolutional filter moves across the input image during the convolution operation.

of the optimization problem. The adequacy of the model's predicted labels are continuously compared to the expected true label thanks to a loss function. This cost function measures a scalar-distance between true values and predicted values. The individual loss is defined for each training pair and is formulated as:

$$l_i = l(f(\mathbf{x_i}; \Theta), \mathbf{y_i}) \tag{2.19}$$

The technical objective behind an optimization procedure is to look for the optimal set of model parameters Θ^* in order to minimize the expected loss, i.e. expected risk. It is denoted as $\mathcal{L}(\Theta)$ and generalizes the individual loss over all samples of the data generating distribution D:

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}}\ \mathcal{L}(\Theta) = \underset{\Theta}{\operatorname{argmin}}\ E_{(x,y)\sim D}\{l_i\} \tag{2.20}$$

The true risk is approximated by the empirical risk $\hat{\mathcal{L}}(\Theta)$ due to the inaccessibility of the complete data generation distribution. The empirical loss is formulated as the average of the individual losses over all training samples:

$$\hat{\mathcal{L}}(\Theta) = \frac{1}{s}\sum_{i=1}^{s} l_i \tag{2.21}$$

The goal of the empirical risk minimisation is to find the optimal parameters of the model such that:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\ \hat{\mathcal{L}}(\Theta) \tag{2.22}$$

Note that Θ^* is most of the time not equal to $\hat{\Theta}$. However, when the dataset size goes to infinity, the difference both values converge to zero.

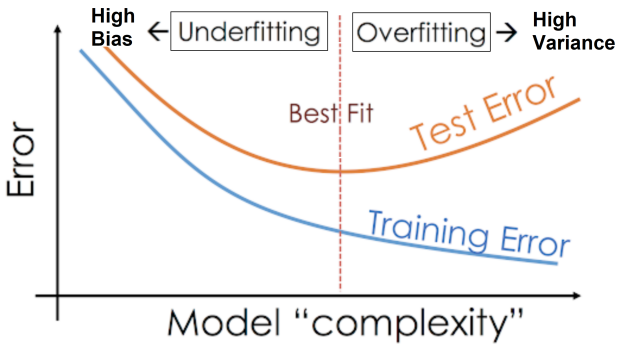


Figure 2.6: Overfitting and underfitting symptoms observed on the train/test Error curve in function of the ML model complexity: underfitting is related to high bias and low variance while overfitting is the result of high variance and low bias.

The choice of the loss function is vital to guarantee better model performances depending on the problem context and the nature of the corresponding dataset. In this PhD manuscript, the research work led to the use of the following prototypical cost functions:

- The Mean Squarred Error: de-facto method for regression problems. It measures the squared Euclidian distane between the model's predicted labels and the targeted true labels:

$$l_{MSE_i} = (f(\mathbf{x_i}, \Theta) - \mathbf{y_i})^2 \tag{2.23}$$

- The Binary Cross-Entropy i.e. the Logarithmic Loss: conventionally applied on classification problems penalizing the model more heavily when its predicted probability diverges from the true binary label using the following equation:

$$l_{BCE_i} = (1 - \mathbf{y_i}) log(1 - f(\mathbf{x_i}; \Theta)) - \mathbf{y_i} log(f(\mathbf{x_i}; \Theta)) \tag{2.24}$$

Generally, we include a regularization term in the loss function to combat Overfitting which occurs when a ML model learns the training data too well, capturing not only the underlying patterns but also the noise and randomness present in the data. Therefore, the model performs exceptionally well on training samples but fails to generalize to new unseen data. The model memorized the training data instead of learning the new underlying relationships. The regularized empirical risk is formulated by adding a penalty term to the loss function, $r(\Theta)$, expressed as:

$$\hat{\mathcal{L}}_r(\Theta) = \frac{1}{s} \sum_{i=1}^{s} l_i + r(\Theta) \tag{2.25}$$

On the other hand, underfitting occurs when a ML model is too simple to capture the underlying patterns in the training data. As a result, it performs poorly in both training and test data. An under-fitted model is unable to learning the complexities

of the data, and its predictions tend to be biased and inaccurate. In figure (2.6), the training/test error curve in function of model complexity states the overfitting where the gap between training and test errors increases, generally explained by high variance symptoms. Contrarily, the underfitting is related to high bias when training and test errors are decreasing. Finally, the balances fit is characterized by low variance and low bias; model complexity and dimensions shouldn't out-pass the corresponding "best fit" limit in order to avoid overfitting.

In summary, overfitting and underfitting are opposite ends of the spectrum in terms of model performance, while regularization is a technique used to strike a balance between the two.

2.4.4 Gradient descent and back-propagation

One of the fundamental algorithms behind training neural networks is Gradient Descent, which enables the optimization of model parameters to minimize the loss function. Gradient Descent iteratively updates the weights and biases of the neural network in the direction of steepest descent of the loss function. This process aims to find the optimal set of parameters that minimizes the difference between the predicted outputs of the network and the true targets. The key idea behind Gradient Descent is to compute the gradients of the loss function with respect to each parameter using the chain rule of calculus. These gradients indicate the direction and magnitude of the steepest ascent of the loss function in the parameter space. By subtracting a fraction of the gradients from the current parameter values, the network gradually moves towards the local or global minimum of the loss function.

Algorithm 1 Gradient Descent

Procedure GD: $\mathcal{S} = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=1}^{s}, f(x;\theta), N_{steps}, \alpha, \theta_0)$
init. $\theta \leftarrow \theta_0$

for $i = 1$ to N_{steps} **do**
Compute gradient $\nabla J(\theta)$ using training data
Update parameters: $\theta = \theta - \alpha \cdot \nabla J(\theta)$
end for
return θ
end GD;

In this algorithm, $f(\mathbf{x_i};\theta)$ is the parametric function, θ represents the model parameters, α is the learning rate (step size), and N_{steps} is the number of iterations. The algorithm iteratively updates the parameters θ by subtracting the product of the learning rate and the gradient of the cost function $J(\theta)$ with respect to the

parameters.

$$\nabla J_\theta(\theta) = \begin{bmatrix} \frac{\partial f(\theta)}{\partial \theta_1} \\ \frac{\partial f(\theta)}{\partial \theta_2} \\ \vdots \\ \frac{\partial f(\theta)}{\partial \theta_n} \end{bmatrix} \tag{2.26}$$

We now consider a mini-batch $\mathcal{B}$ of size b randomly containing samples from the training set $\mathcal{S}$ of size s.

Algorithm 2 Gradient Descent using mini-batch

Procedure MBGD: $(\mathcal{S} = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=1}^{s}, f(x;\theta), N_{steps}, b, \alpha, \theta_0))$
init. $\theta \leftarrow \theta_0$

for $i = 1$ to N_{steps} **do**
S $\leftarrow Shuffle(\mathcal{S})$
$\mathcal{B} \leftarrow \mathcal{S}[0:b]$
$Computegradient \nabla J(\theta)$ using training data
Update parameters: $\theta = \theta - \alpha \cdot \nabla J(\frac{1}{b}\sum_{(\mathbf{x_i},\mathbf{y_i})\in\mathcal{B}} l(f(\mathbf{x_i};\theta), \mathbf{y_i}))$
end for
return θ
end MBGD ;

Gradient descent with mini-batch involves updating model parameters using a subset of training data in each iteration, which helps in faster convergence and efficient memory usage. On the other hand, gradient descent with momentum incorporates a moving average of past gradients to achieve faster convergence by dampening oscillations and enhancing gradient directions, resulting in improved training efficiency and stability.

Algorithm 3 Gradient Descent using momentum

Procedure MGD: $\mathcal{S} = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=1}^{s}, f(x;\theta), N_{steps}, m, \alpha, \theta_0)$
init. $\theta \leftarrow \theta_0$
$m \leftarrow 0$

for $i = 1$ to N_{steps} **do**
Compute gradient $\nabla J(\theta)$ using training data
$m \leftarrow \beta m + (1-\beta)\nabla_\theta(\frac{1}{s}\sum_{(\mathbf{x_i},\mathbf{y_i})\in\mathcal{S}} l(f(\mathbf{x_i},\theta), \mathbf{y_i}))$
Update parameters: $\theta = \theta - \alpha m$
end for
return θ
end MGD;

Alternatively, backpropagation is an efficient algorithm used to compute the gradients required by Gradient Descent. It leverages the chain rule of calculus to

efficiently propagate the errors from the output layer back to the network's earlier layers. In the forward pass, the inputs are fed through the network, and the outputs are computed layer by layer. During the backward pass, the gradients of the loss function with respect to the outputs are first computed. Then, these gradients are successively back-propagated through the layers, allowing the computation of the gradients with respect to the weights and biases. The backpropagation algorithm efficiently calculates these gradients by reusing intermediate results obtained during the forward pass, avoiding redundant computations. By utilizing these gradients, the weights and biases of the neural network are updated iteratively using Gradient Descent, ultimately leading to the convergence of the network towards an optimal solution. For instance, let f, g and h be three real functions where $y = f(x), z = g(f(x)), g(y)$ and $w = h(g(f(x)) = h(g(y)) = h(z)$, modeling the input-output functions of 3 layers in a neural network. In order to calculate the derivative of the output of the last layer w with regard to the input x, we rely on the following equation:

$$\frac{\partial w}{\partial x} = \frac{\partial w}{\partial z} \cdot \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x} \tag{2.27}$$

Using the Jacobian matrix in backpropagation offers several advantages in training neural networks enabling efficient computation of gradients for each layer's inputs which generally leads to faster and more accurate updates to the model's parameters. For example, let $y = f(x)$ where $f : \mathbb{R}^n \to \mathbb{R}^m$. The Jacobian matrix is expressed as:

$$J_x f(x) = \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{2.28}$$

The Jacobian matrix simplifies the implementation of complex architectures, as it encapsulates the partial derivatives of each layer's output with respect to all inputs, reducing the risk of errors in the gradient calculations during backpropagation.

The combination of Gradient Descent and Backpropagation forms the basis of training deep neural networks. This iterative optimization process, guided by the computed gradients, allows neural networks to automatically learn and adapt their parameters to minimize the loss function. With the advent of powerful computational resources and the development of efficient optimization techniques, deep neural networks can now be trained on large-scale datasets, enabling the extraction of meaningful representations from complex data [14] [18] [20]. Gradient Descent and Backpropagation have played a pivotal role in the success of deep learning, allowing for breakthroughs in various fields, including computer vision, natural language processing, and speech recognition. These algorithms have also been instrumental in addressing challenges in wireless communications, such as beam alignment, channel estimation, and interference mitigation, enabling the optimization and enhancement of future mmWave massive MIMO systems.

2.5 Learning paradigms and optimization problems

Artificial Intelligence and computer science have witnessed remarkable advancements over the years, revolutionizing various fields and shaping the way we interact with technology. From the early development of formal neurons and the exploration of neural architectures, to the advent of gradient descent and back-propagation algorithms, AI has evolved into a powerful tool for solving complex problems. These advancements have paved the way for the emergence of multiple machine learning paradigms, including supervised, unsupervised, and reinforcement learning in addition to several optimization problems, such as linear programming, quadratic programming, nonlinear programming, integer programming..

2.5.1 Supervised, Unsupervised and Reinforcement Learning

We distinguish three fundamental paradigms in machine learning:

- Supervised learning where the model learns from labeled training data, based on a training set $\mathcal{S} = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=1}^{s}$ where each input $\mathbf{x_i}$ is associated with a corresponding output $\mathbf{y_i}$. The goal is to learn a mapping function $f(\mathbf{x_i}, \mathbf{y_i})$ that can generalize to unseen inputs and produce accurate predictions or classifications. Supervised learning can be formulated as an optimization problem, where the objective is to minimize the discrepancy between the predicted outputs and the true labels.

- Unsupervised learning, on the other hand, deals with unlabeled data. The goal is to discover the underlying structure, patterns, or relationships in the data without explicit guidance. Clustering and dimensionality reduction are common tasks in unsupervised learning. Clustering aims to group similar instances together, while dimensionality reduction techniques aim to reduce the dimensionality of the data while preserving important information. Unsupervised learning algorithms leverage statistical properties and patterns in the data to extract meaningful representations using PCA, k-NN, Kohenen Map and t-SNE approaches.

- Reinforcement learning takes a different approach by focusing on learning optimal decision-making policies through interactions with an environment. In this setting, an agent learns to take actions in an environment to maximize a cumulative reward signal. The agent receives feedback in the form of rewards or punishments based on its actions, which guides its learning process. Reinforcement learning often involves Markov decision processes and utilizes techniques such as value iteration, policy iteration, or Q-learning to find the optimal policy.

These three learning paradigms have distinct characteristics and applications. Supervised learning is commonly used in tasks such as image classification, speech recognition, and natural language processing. Unsupervised learning finds applications in anomaly detection, data clustering, and generative modeling. Reinforcement learning is well-suited for problems with sequential decision-making, such as robotics, game playing, and autonomous systems.

In the context of wireless communication systems, these learning paradigms find various applications. For instance, supervised learning techniques [12] [17] can be used to optimize beamforming in mmWave massive MIMO systems by learning the optimal beamforming weights based on labeled channel state information. Unsupervised learning methods can be employed for interference detection and mitigation in wireless networks, where the goal is to identify and separate different sources of interference. Reinforcement learning can be utilized for dynamic resource allocation, where the system learns to allocate radio resources effectively to maximize network performance while considering changing traffic conditions and channel conditions [18] [19].

Overall, the integration of machine learning techniques, including supervised, unsupervised, and reinforcement learning, offers promising avenues for advancing wireless communication systems by enabling intelligent decision-making, optimization, and adaptation.

2.5.2 Families of optimization problems

Optimization problems involve finding the best solution among a set of feasible options based on a defined objective function and a set of constraints. Optimization problems can be classified into two families: convex and non-convex, each having sub-families, based on their characteristics and mathematical formulations (discrete/continuous, linear/non-linear):

- Linear programming where the objective function and constraints are linear. LP has been extensively studied and finds applications in resource allocation, production planning, and portfolio optimization [36]. The simplex method and interior-point methods are widely used to solve LP problems efficiently.

- Quadratic programming is another family of optimization problems that involves a quadratic objective function subject to linear constraints. QP problems arise in various fields such as robotics, control systems, and support vector machines . Specialized algorithms, including the active set method and interior-point methods, are employed to solve QP problems effectively.

- Nonlinear programming deals with optimization problems where the objective function or constraints are nonlinear. NLP encompasses a broad range of problems, from smooth (continuously differentiable objective function, with no abrupt changes or discontinuities) and convex optimization (where the function's graph forms a bowl-like shape, allowing for efficient algorithms to find the global optimum and guaranteeing that any local optimum found is also the

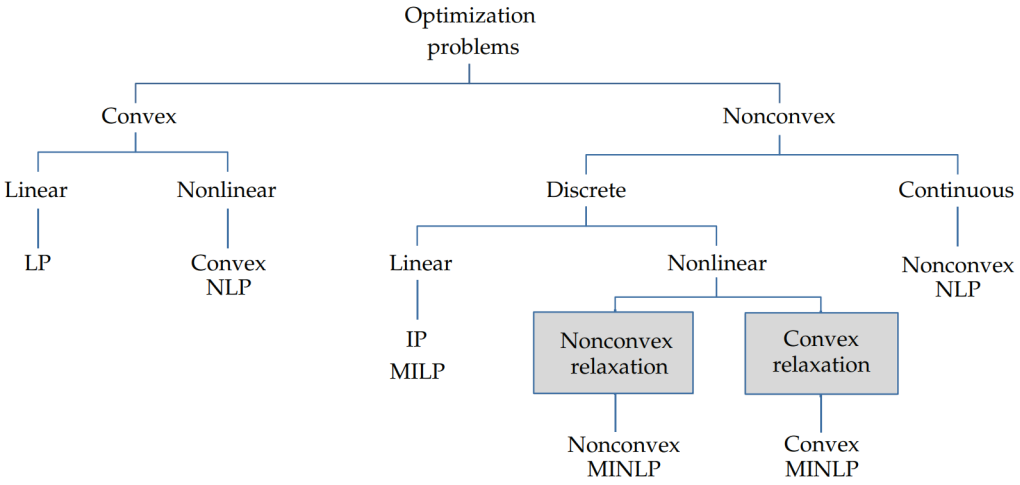


Figure 2.7: Families of Optimization problems [37]

global one), to highly non-convex and non-smooth optimization. Techniques like gradient-based methods, such as gradient descent and Newton's method, as well as metaheuristic algorithms like genetic algorithms and simulated annealing, are used to solve NLP problems.

- Integer programming and its subset, mixed-integer programming. In IP problems, one or more variables are constrained to take only integer values, while in MIP problems, some variables are integers, and others are continuous. IP and MIP have applications in diverse areas, including logistics, scheduling, and network design. Branch-and-bound, cutting plane, and branch-and-cut algorithms are commonly employed to solve IP and MIP problems.

Convex optimization focuses on problems where the objective function and constraints are convex. Convex optimization problems have attractive mathematical properties, allowing for efficient and guaranteed global optimization. Interior-point methods, proximal gradient methods, and augmented Lagrangian methods are frequently used to solve convex optimization problems.

These families of optimization problems, resumed in (2.7), provide a framework for modeling and solving a wide range of real-world challenges. Throughout this PhD, the non-convex problems are investigated. In the context of machine learning, optimization plays a crucial role in training models, estimating parameters, and solving inference problems. By formulating machine learning tasks as optimization problems, researchers and practitioners can leverage the rich literature of optimization algorithms to find optimal solutions and drive advancements in AI and wireless communication systems.

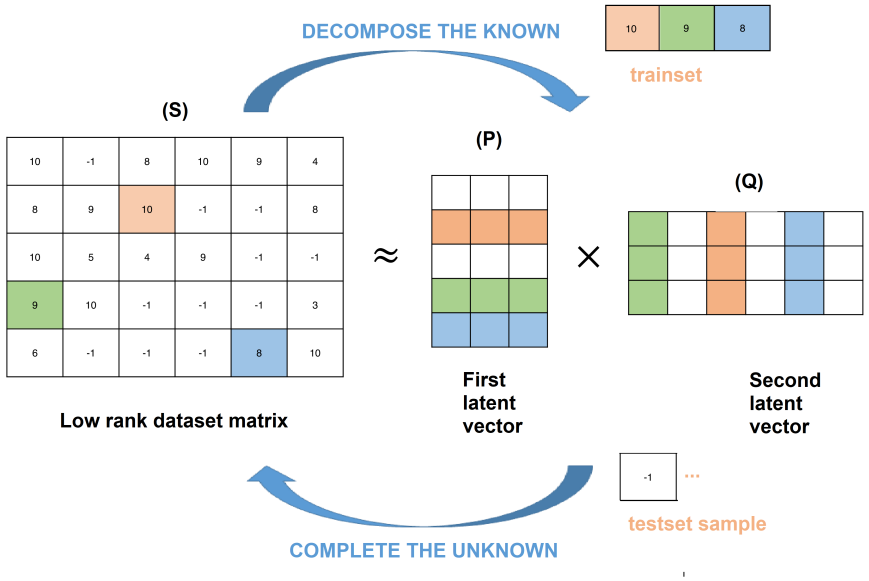


Figure 2.8: Matrix Factorization simplified diagram representation for 6×5 input matrix: $\mathbf{S}$ is first decomposed as the product of two latent factors (model's parameters $\mathbf{P}$ and $\mathbf{Q}$) in order to fill the unknown coefficients

2.6 Matrix Factorization for low-rank matrix completion

Matrix factorization is a popular technique in machine learning and data analysis that aims to decompose a given matrix into two or more lower-rank matrices. The goal is to approximate the original matrix by finding a reduced-dimensional representation that captures its underlying structure and latent features. Matrix factorization has been widely applied in various domains, including recommender systems, collaborative filtering, and dimensionality reduction. By leveraging linear algebra and optimization methods, matrix factorization provides a powerful tool for extracting meaningful information from high-dimensional data and enabling efficient computation on large-scale datasets.

Low-rank matrix completion is a fundamental problem in the field of machine learning and data analysis. It deals with the task of recovering a complete matrix from a limited set of observed entries, where the underlying matrix is assumed to have a low-rank structure. The motivation behind low-rank matrix completion arises from the fact that many real-world datasets are inherently incomplete or corrupted, and being able to accurately recover the missing entries is crucial for various applications. The main idea behind low-rank matrix completion is to exploit the inherent low-rank structure of the matrix to estimate the missing entries. The low-rank assumption assumes that the rank of the underlying matrix is much smaller than its dimensions, indicating that the matrix can be well-approximated by a matrix of significantly lower rank. This assumption is often valid in practice, as many real-world datasets exhibit inherent redundancy and can be effectively represented in a

lower-dimensional space.

To solve the low-rank matrix completion problem, various algorithms and techniques have been developed. In recommender systems, for example, it can be used to predict user preferences or missing ratings in a user-item rating matrix [33]. In image and video processing, it can be employed for inpainting missing regions or reconstructing corrupted images. Additionally, it finds applications in collaborative filtering, data imputation, and network inference, among others. The development of efficient and accurate algorithms for low-rank matrix completion continues to be an active area of research, aiming to advance the field of machine learning and enhance the capabilities of data analysis and decision-making processes.

The low-rank matrix completion is formulated in this PhD as a non-convex optimization problem, where the objective is to find a low-rank matrix that fits the observed entries while satisfying certain constraints. These methods often employ iterative algorithms that aim to minimize a non-convex objective function, such as the sum of squared errors or the weighted nuclear norm. Although finding the global minimum of a non-convex optimization problem is generally challenging, various heuristics and optimization techniques, such as alternating minimization or gradient descent, have been developed to effectively explore the solution space and converge to promising solutions. Non-convex formulations of low-rank matrix completion offer a flexible and powerful framework for handling complex data structures and can lead to improved accuracy in matrix recovery tasks.

The goal of low-rank MF method is to learn from train samples and continuously optimize the latent vectors of dimension D, $\{\mathbf{p_u}, \mathbf{q_i}\}_{(u,i)\in\mathcal{K}}$. The training samples are the known coefficients of our dataset matrix $\mathbf{S}$, which holds the sounded Received Signal Energies for training beam pairs. The dataset is then split into a training set, denoted $\mathcal{K}$ and a test set, denoted $\mathcal{L}$, the unknown (non-sounded) samples that we aim to complete using MF. Recall that for MF, the latent vectors satisfy $\mathbf{p_u} \in \mathbb{R}^D, \mathbf{q_i} \in \mathbb{R}^D, \forall(u,i) \in$. The loss function is then formulated as:

$$f((\mathbf{p_u}, \mathbf{q_i})_{(u,i)\in\mathcal{K}}) = \sum_{(u,i)\in\mathcal{K}} \left[\frac{1}{|\mathcal{K}|} \left([\mathbf{S}]_{u,i} - \mathbf{p_u}^T \mathbf{q_i} \right)^2 + \lambda_i \|\mathbf{q_i}\|_2^2 + \mu_u \|\mathbf{p_u}\|_2^2 \right] \quad (2.29)$$

Therefore, the optimization Problem for MF is expressed as:

$$(P_{MF}) : \{\widehat{\mathbf{p_u}}, \widehat{\mathbf{q_i}}\} \quad \begin{cases} \underset{\{\mathbf{p_u}, \mathbf{q_i}\}_{(u,i)\in\mathcal{K}}}{argmin} & f((\mathbf{p_u}, \mathbf{q_i})_{(u,i)\in\mathcal{K}}) \\ \mathbf{p_u} \in \mathbb{R}^D, \ \mathbf{q_i} \in \mathbb{R}^D \end{cases}$$

In (P_{MF}), the Received Signal Energies of the sounded beam-pairs are known, i.e., training set, $\{[\mathbf{S}]_{u,i} | \forall(u,i) \in \mathcal{K}\}$, and the optimization variables that are needed to be learned are the latent factors corresponding the training set, $\{\mathbf{q}_i, \mathbf{p}_u | \forall(u,i) \in \mathcal{K}\}$. The optimal latent vectors are denoted as $\{\widehat{\mathbf{p}}_u, \widehat{q}_i\}_{(u,i)\in\mathcal{K}}$.

We resolve the MF problem (P_{MF}) using the following methods:

- Block Coordinate Descent often denoted as Alternating Least Squares.
- BCD with Stochastic Gradient Descent.

- Block Gradient Descent, that merges BCD and Gradient Descent definitions.

Our proposed NMF follows the exact steps as MF, with the main difference of constraining the latent vectors to be non-negative:

$$\mathbf{p}_u \in \mathbb{R}_+^D \ , \ \mathbf{q}_i \in \mathbb{R}_+^D, \forall \ (u,i) \in \mathcal{K} \tag{2.30}$$

Given that, the loss for NMF is the regularized empirical risk in (P_{NMF}) with non-negative parameters:

$$(P_{NMF}) : \ \{\widehat{\mathbf{p_u}}, \widehat{\mathbf{q_i}}\} \quad \begin{cases} \underset{\{\mathbf{p_u}, \mathbf{q_i}\}_{(u,i)\in\mathcal{K}}}{argmin} \quad f((\mathbf{p_u}, \mathbf{q_i})_{(u,i)\in\mathcal{K}} \\ \mathbf{p_u} \geq \mathbf{0} \ , \ \mathbf{q_i} \geq \mathbf{0} \end{cases} ,$$

where $\mathbf{0}$ is the all-zero vector of dimension D. Likewise, we solve the NMF problem, (P_{NMF}), using BCD, SGD, and BGD.

2.7 Conclusion

Chapter 2 provides a comprehensive introduction to ML techniques and paradigms. It covers the historical breakthroughs and key developments in AI then illustrates a technical overview for Statistical Learning and Deep Learning. Various Learning paradigms and optimization problems are discussed. In the next chapter, the focus shifts to Beam Alignment in massive mmWave MIMO systems, highlighting the unique challenges and opportunities associated with this field.

Chapter 3

Overview of Beam Alignment for mmWave MIMO communications

"It is through science that we prove, but through intuition that we discover."

Henri Poincare.

3.1 Introduction

Beam alignment refers to the process of aligning the transmitting and receiving beams between the base station and user equipment, ensuring optimal signal strength and minimizing interference. Therefore, it plays a crucial role in mmWave MIMO systems, where the use of directional beams is necessary to establish reliable communication links. In these high-frequency bands, narrow beams are employed to combat severe path loss and enhance system capacity. This procedure is essential to exploit the directional characteristics of mmWave channels, maximize spatial multiplexing gain, and achieve high data rates in future wireless communication systems. In this chapter, we present the terminology associated with this technical issue, taking into account the characteristics of mmWave band propagation, its constraints, and the technical obstacles it poses. Simultaneously, we give an overview of the signal processing methods employed to address the suite of Beam Management challenges, encompassing Beamforming, Beam Alignment, Beam Sweeping, and Beam Tracking.

3.2 The mmWave band and propagation properties

Millimeter-wave MIMO refers to the use of multiple antennas at both the transmitter and receiver in wireless communication systems operating in the millimeter-wave frequency range. It harnesses the abundant spectrum available in these high-frequency bands to enable both high-capacity and high-speed data transmission, and offers

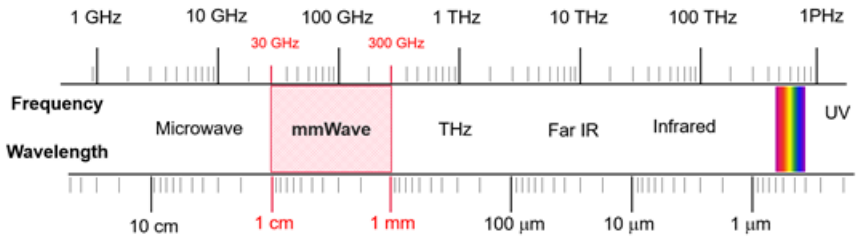


Figure 3.1: mmWave spectrum: the band of frequencies between 30 and 300 GHz [38]

significant potential for meeting the ever-increasing demands of future wireless networks.

3.2.1 mmWave spectrum

The mmWave spectrum in (3.1) refers to a portion of the electromagnetic spectrum that spans frequencies between 30 and 300 GHz. This frequency range is significantly higher than the traditional sub-6 GHz bands used in most wireless communication systems. The mmWave spectrum offers several advantages, such as a vast amount of available bandwidth and the ability to support high data rates:

- 30-300 GHz (E-band): The E-band, spanning 30 to 300 GHz, is a millimeter-wave range well-suited for microwave and millimeter-wave communication systems. Its high-frequency attributes facilitate robust point-to-point data transmission, notably in 5G backhaul links and dense urban environments.
- 57-71 GHz (V-band): Within the V-band, operating between 57 and 71 GHz, mmWave frequencies find applicability in various wireless communication scenarios, particularly as a component of short-range, high-capacity connections such as small cell deployments and fixed wireless access.
- 71-76 GHz and 81-86 GHz (W-band): The W-band, encompassing 71-76 GHz and 81-86 GHz, serves as a pivotal region for advanced radar, spectroscopy, and scientific research due to its high-frequency attributes. It enables fine-resolution radar imaging, atmospheric sensing, and investigations into molecular spectral lines.
- 140-220 GHz (D-band): Extending from 140 to 220 GHz, the D-band plays a crucial role in atmospheric studies, Earth observation, and specialized communication systems. It is notably adept at penetrating adverse weather conditions and supports applications like weather radar and environmental monitoring.
- 220-325 GHz (F-band): The F-band, ranging from 220 to 325 GHz, serves as a cornerstone in radio astronomy, spectroscopy, and space research endeavors. Its capacity to detect precise molecular transitions makes it instrumental in probing celestial objects and examining the universe's composition.

- 325-500 GHz (G-band): Operating within the 325-500 GHz range, the G-band holds significance in scientific exploration, including radio astronomy and space communications. This frequency regime is integral for studying cosmic microwave background radiation and establishing high-frequency space communication links.

Furthermore, the use of mmWave frequencies enables the deployment of highly directional beamforming techniques, leveraging the large antenna arrays at both the transmitter and receiver to focus the energy in specific directions. This enables the possibility of achieving significantly higher data rates and supporting massive connectivity in dense network scenarios. Despite the challenges associated with mmWave propagation, advancements in antenna design, signal processing, and beamforming techniques are enabling the realization of mmWave communication systems with improved performance and reliability.

3.2.2 mmWave limitations

Path Loss and Penetration Loss: One of the primary challenges of mmWave communication is increased path loss and reduced penetration through obstacles. The higher frequency signals are more susceptible to attenuation and absorption by atmospheric gases, rain, foliage, and buildings. This results in shorter communication range and limited signal penetration, requiring careful consideration of the deployment and placement of mmWave devices according to [39]. These limitations and challenges behind the mmWave technologies are resumed in (3.2):

- Blockage and Line-of-Sight Requirement: mmWave signals are highly directional and sensitive to blockages. Even small obstructions, such as buildings or human bodies, can cause significant signal attenuation. Maintaining a clear line-of-sight between the transmitter and receiver becomes crucial for reliable mmWave communication. The presence of blockages can lead to frequent signal interruptions and reduced coverage, requiring the development of effective beamforming and beam tracking mechanisms.

- Limited Coverage Area: Due to the high path loss and susceptibility to blockage, mmWave signals have a limited coverage area. The effective coverage range of mmWave base stations and devices is typically shorter compared to lower frequency bands. This limitation necessitates the deployment of a dense network infrastructure with small cell sizes to ensure reliable connectivity and seamless handoffs between cells.

- Mobility and Doppler Effects: The mobility of devices introduces additional challenges in mmWave communication. As devices move, there are rapid changes in the channel conditions, leading to significant Doppler shifts. The high frequency nature of mmWave signals amplifies the Doppler effects, resulting in frequency offsets and potential signal distortion. Robust mechanisms for beam tracking and adaptation to fast-changing channel conditions are required to maintain reliable communication in mobile scenarios.

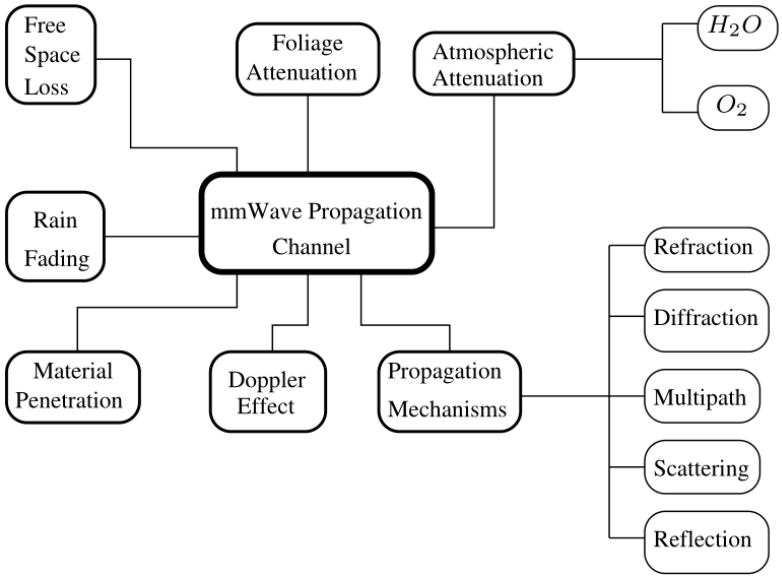


Figure 3.2: Overview of mmWave propagation limitations and challenges [39]

- Limited Diffraction and Scattering: mmWave signals have limited ability to diffract and scatter around obstacles compared to lower frequency signals. This limits the ability to bypass obstacles and reach non-line-of-sight locations. The reliance on LOS communication further exacerbates the challenges of mmWave propagation, necessitating the development of advanced beamforming techniques to steer and focus the signals towards the desired receivers.

- Higher Sensitivity to Noise and Interference: The higher frequency bands used in mmWave communication are more susceptible to noise and interference, which can degrade the quality of the received signal. Background noise, interference from other devices, and multipath effects can impact the signal quality and overall system performance. Robust interference mitigation and noise suppression techniques are essential for maintaining reliable and efficient mmWave communication.

Addressing all mmWave limitations [39] requires a combination of advanced signal processing techniques, adaptive beamforming, intelligent resource allocation, and sophisticated network planning. Overcoming these challenges will enable the realization of the full potential of mmWave communication systems in delivering high-capacity, low-latency, and ultra-reliable wireless connectivity for future applications.

3.2.3 mmWave massive MIMO challenges

Massive Multiple-Input Multiple-Output refers to a wireless communication system that utilizes a large number of antennas at both the transmitter and receiver ends. This technology leverages the spatial dimension to enhance system performance by simultaneously transmitting multiple data streams to multiple users. With the

ability to spatially separate signals and mitigate interference, massive MIMO offers significant gains in terms of capacity, spectral efficiency, and reliability, making it a promising solution for next-generation wireless networks. The large available bandwidth in mmWave bands enables high data rates and supports the massive connectivity requirements of modern applications. However, as mentioned in the previous subsection and [39], mmWave signals are susceptible to higher path loss, attenuation, and sensitivity to blockages due to their shorter wavelengths. In addition, we define the mmWave Coherence Interval as the time where the channel coefficients remain static. For mmWave, this critical interval is so short and the implementation of Beam Alignment techniques are often tackled with this enormous practical constraint [40]. This introduces challenges in maintaining reliable and robust communication links, particularly in massive MIMO systems with a large number of antennas. In the context of beam management techniques, mmWave massive MIMO plays a crucial role in optimizing beamforming, beam alignment, and beam tracking. The beam management techniques of the next subsection are fundamental for realizing the full potential of mmWave massive MIMO systems, enabling high-capacity and reliable wireless communication in challenging propagation environments.

3.3 Overview of Beam Management techniques for MIMO systems

3.3.1 Beamforming, precoding and combining

Beamforming, precoding and combining are sophisticated signal processing methods applied in wireless communication systems, strategically improving signal transmission and reception performance:

- Beamforming is a signal processing technique used in wireless communication systems to enhance the transmission and reception of signals. It involves manipulating the amplitude and phase of the transmitted or received signals to concentrate the signal power in a specific direction, known as the beam. By steering the beam towards the intended receiver, beamforming improves signal strength, increases coverage, and reduces interference, thereby enhancing the overall system performance. In figure (3.3), the increase in the number of antennas helps obtaining more beamforming directivity where the main lobe gets narrower and strictly directed to the target. In this PhD, we propose to focus on massive MIMO systems for future generations where hundreds and thousands of antennas are used in order to maximise the beam forming gain by selecting the constructive interference between beams and filtering the destructive one. Commonly used mathematical methods include Zero Forcing [41], Maximum Likelihood and Minimum Mean Squared Error Beamforming [42].

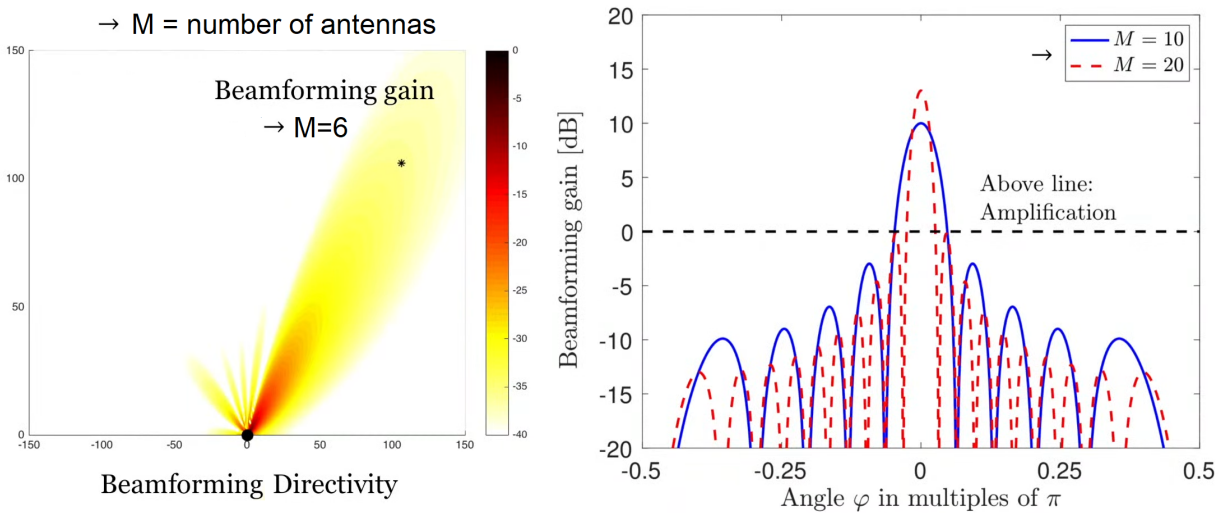


Figure 3.3: Beamforming gain and directivity in MIMO systems: more antennas gives narrower and more directive lobes [40]

- Precoding is a technique used in multi-antenna systems to optimize the transmitted signal based on the channel conditions. Mathematically, it is a generalization of beamforming. It involves applying specific linear transformations to the data symbols before transmission, taking into account the channel state information. Precoding enables the transmitter to exploit the channel characteristics to enhance the received signal quality at the receiver. By premultiplying the data symbols with the precoding matrix, the transmitter can shape the transmitted signal to maximize the signal-to-interference-plus-noise ratio at the receiver. ZF Precoding and MMSE precoding are widely used approaches in the literature [42].

- Precoding and Combining, also known as receive beamforming, is the counterpart of beamforming at the receiver side. It involves combining the received signals from multiple antennas in a way that optimally combines the signals to improve the received signal quality. By adjusting the combining weights based on the channel state information, combining techniques enhance the desired signal while suppressing interference and noise. Combining can significantly improve the overall system performance by mitigating the effects of fading, interference, and noise in the received signal. Frequently employed Combining methods include Maximum Ratio Combining [42], Selection Combining [43] and Singular Value Decomposition [44] and [45].

These techniques play crucial roles in optimizing the performance of wireless communication systems, especially in scenarios with challenging channel conditions, interference, and multi-path effects.

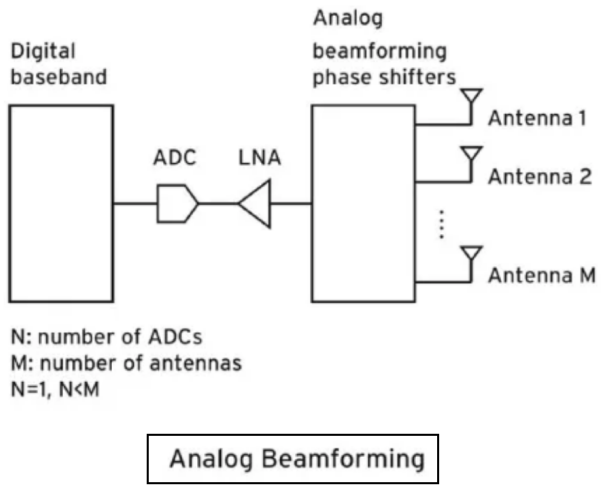


Figure 3.4: Simplified fully-analog beamforming architecture

3.3.2 Analog, digital and hybrid MIMO architectures

The design of MIMO systems in terms of architectural requirements is one of the most important challenges in Beam Management problems. We distinguish three main architectures: fully-analog, fully-digital or hybrid involving both.

Fully-analog MIMO architectures are characterized by using analog components throughout the entire signal processing chain. The key advantage of fully-analog architectures is their simplicity and low implementation complexity. Since the signals remain in the analog domain, they can be processed using highly efficient and low-power analog circuitry. However, fully-analog architectures suffer from limitations in terms of flexibility and adaptability. Any changes or updates to the signal processing algorithms require modifications to the analog circuitry, which can be time-consuming and costly. Additionally, fully-analog architectures may be susceptible to interference and noise, as there are no digital signal processing techniques available to mitigate these effects. Figure (3.4) provides a simplified diagram representation of a fully-analog architecture.

On the other hand, fully-digital MIMO architectures rely on digital signal processing techniques for all stages of signal transmission and reception. The incoming signals are first converted from analog to digital domain, and all subsequent processing is performed digitally. Fully-digital architectures also offer superior interference and noise mitigation capabilities, thanks to the powerful digital processing algorithms and techniques. However, fully-digital architectures can be more complex and require higher computational resources compared to their analog counterparts. The need for high-speed and high-resolution analog-to-digital converters and digital-to-analog converters adds to the overall cost and power consumption of the system. Figure (3.5) provides a simplified diagram representation of a fully-digital architecture.

Hybrid architectures, in figure (3.6), combine the advantages of both fully-analog and fully-digital approaches. They employ a combination of analog and digital com-

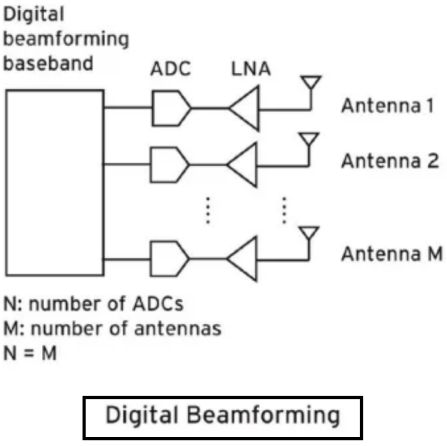


Figure 3.5: Simplified fully-digital beamforming architecture

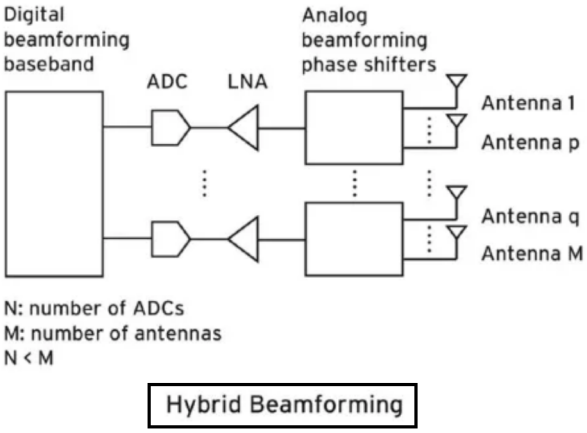


Figure 3.6: Simplified analog-digital beamforming architecture

ponents in the signal processing chain. Typically, the RF front-end and the initial stages of signal processing are implemented in the analog domain, while the later stages involve digital processing. This allows for a balance between performance, flexibility, and complexity. Hybrid architectures can leverage the efficiency of analog processing for initial RF tasks while benefiting from the flexibility and adaptability of digital processing for more advanced signal processing functions. However, the design and optimization of hybrid architectures require careful consideration of the trade-offs between analog and digital components, as well as the interfaces between them. Efficient coordination and synchronization between analog and digital processing stages are essential to ensure optimal system performance.

3.3.3 Beam Alignment

Beam Alignment is a fundamental operation in mmWave MIMO communication systems, which aims to establish an optimal beamforming link between the transmitter and the receiver, as shown in figure (3.7) and (3.8). It involves the process of

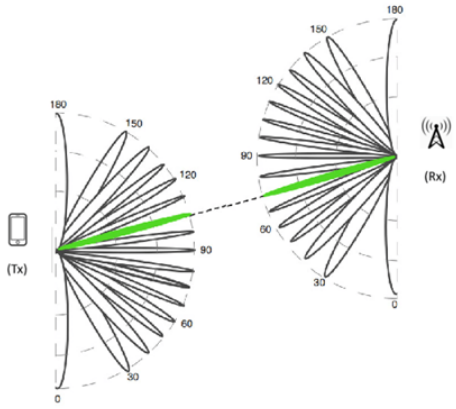


Figure 3.7: Beam Alignment technical objective: accurately direct the beams between UE/BS using codebooks holding beam patterns for each antenna pair in both sides of the transmission

aligning the transmit and receive beams in the most favorable direction to maximize the signal quality and improve system performance. Beam Alignment is crucial in mmWave systems due to the highly directional nature of mmWave signals, where precise beam steering is required to overcome the severe path loss and limited diffraction characteristics of these high-frequency signals, stated in the previous section of this chapter.

During the Beam Alignment process, the transmitter and receiver exchange control information to estimate the channel conditions and determine the optimal beamforming vectors. Beam sweeping is often employed as an initial step, where a predefined set of beamforming vectors is systematically swept across the angular space to explore potential beam directions. The receiver then measures the received signal quality for each beam direction and feeds this information back to the transmitter. Based on the feedback, the transmitter selects the beamforming vector that maximizes the received signal strength or other performance metrics. This process is typically iterated until an optimal beam alignment is achieved.

Beam Alignment is often denoted Beam Training in the literature in reference to the training required to accurately direct the beams, where this literature survey in [1] resumes in three families, illustrated in figure (3.10).

Codebook-based Beam Alignment:
In codebook-based Beam Alignment, predefined sets of beamforming vectors, known as codebooks, are employed at both the transmitter and receiver to establish a reliable communication link. These codebooks consist of distinct beamforming patterns (AoD, AoA..) that facilitate efficient beam alignment in millimeter-wave frequencies, allowing for optimal signal transmission between transmitters and receivers.

- **Discrete Fourier Transform codebook**: utilizes the mathematical princi-

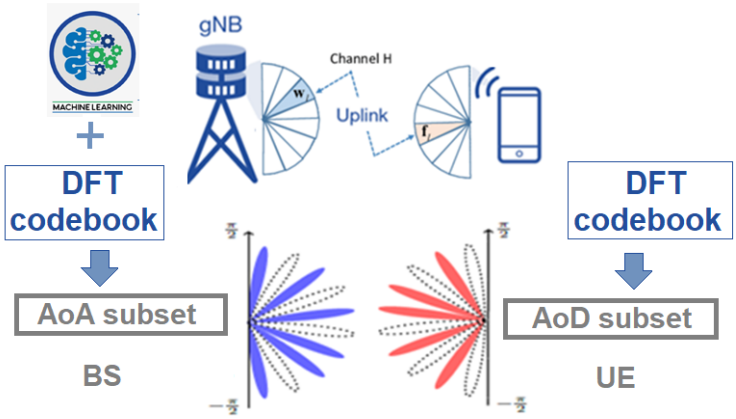


Figure 3.8: Simplified illustration of an Uplink scenario for Beam Alignment using (AoA, AoD) from UE/BS DFT codebooks

ples of the DFT to create a set of orthogonal beamforming vectors:

$$\mathcal{T}_{\text{DFT}} = \left\{ \frac{1}{\sqrt{N}} \cdot e^{-j\frac{2\pi}{N}\cdot k\cdot n} \,\middle|\, 0 \leq k \leq N-1,\, 0 \leq n \leq N-1 \right\} \tag{3.1}$$

where k and n are the indices indicating the beams in the codebook, N represents the number of antennas in the array, and $e^{-j\frac{2\pi}{N}\cdot k\cdot n}$ calculates the complex exponential values for the beamforming elements. The normalization factor $\frac{1}{\sqrt{N}}$ ensures that the beamforming vectors have unit norm. The resulted beams from the DFT codebooks are equispaced.

- **Uniform Linear Array codebook:** Well known for its simplicity, ULA codebook utilizes evenly spaced antenna elements to cover a specific angular range, enabling beamforming in both horizontal and vertical dimensions.

- **Uniform Planar Array codebook:** extends beamforming capabilities to a two-dimensional plane, employing a grid of antennas in both horizontal and vertical directions. This type of codebook enhances spatial coverage and diversity, suitable for complex mmWave environments with diverse user locations.

- **Non-Uniform Linear Array codebook:** Unlike Uniform codebooks, the Non-Uniform Linear Array codebooks employ unequally spaced antenna elements, allowing for customized beamforming patterns tailored to specific angular regions. This adaptability enhances beamforming precision, making it suitable for scenarios where focusing on specific angles is crucial.

- **Singular Value Decomposition based codebook:** can be represented as follows:

$$\mathbf{F}_{\text{SVD}} = \mathbf{U}_{\text{TX}} \cdot \mathbf{V}_{\text{RX}}^{H}. \tag{3.2}$$

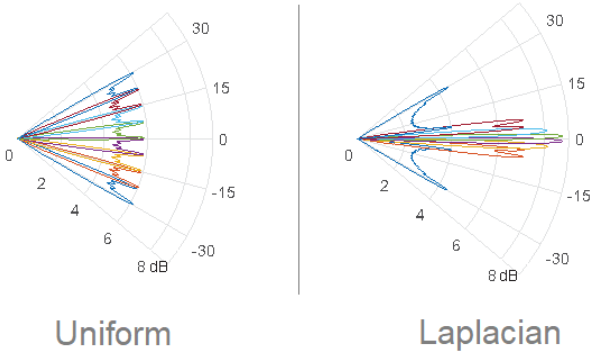


Figure 3.9: Uniform codebook beams vs Laplacian codebook beams [46]

Here, $\mathbf{U}_{TX}, \mathbf{V}_{RX}^{H}$ are the unitary matrices obtained from the singular value decomposition of the channel matrix $\mathbf{H}$, formulated as:

$$\mathbf{H} = \mathbf{U}_{TX} \cdot \mathbf{\Sigma} \cdot \mathbf{V}_{RX}^{H} \tag{3.3}$$

with Σ being a diagonal matrix containing the singular values of $\mathbf{H}$.

- **Laplacian codebook:** its distinctive characteristic lies in utilizing phase shifts with Laplacian-distributed angles, ϕ, as follows [46]:

$$f(\phi) = \frac{k}{\sqrt{2}\sigma_{\phi}} exp(-\frac{\sqrt{2}|\phi - m_{\phi}|}{\sigma_{\phi}}), \phi \in [\phi_{min}, \phi_{max}] \tag{3.4}$$

where m_{ϕ} denotes the mean of the angles and σ_{ϕ} is the standard deviation. In figure (3.9), we compare the resulted beams from Uniform codebooks and Laplacian codebooks using 8 antennas. In summary, the diverse array of codebooks, including the classic DFT, the evenly distributed Uniform codebook, the SVD-based codebook, and the Laplacian codebook, collectively enrich the toolkit for mmWave MIMO beam alignment.

3.3.4 SotA Beam Alignment and benchmark

As we mentioned in the first chapter, the literature survey illustrates two families of Beam Alignment approaches: classical BA and ML based BA. Classical Beam Alignment techniques tend to use more and more structured Beam Alignment design such as hierarchical multi-level codebooks in [5] where training beamforming vectors are constructed with different beam widths at different levels, overlapped beam pattern in [6] where the main idea is to augment the amount of information carried by each channel measurement reducing the required channel estimation time, beam coding in [7] where we assign a unique code-signature to each beam angle in addition to subspace estimation/decomposition based BA in [8]. Compressed sensing-based algorithms in [9] is also used in this context taking advantage of channel sparsity. The majority of prior work using compressed sensing was limited to

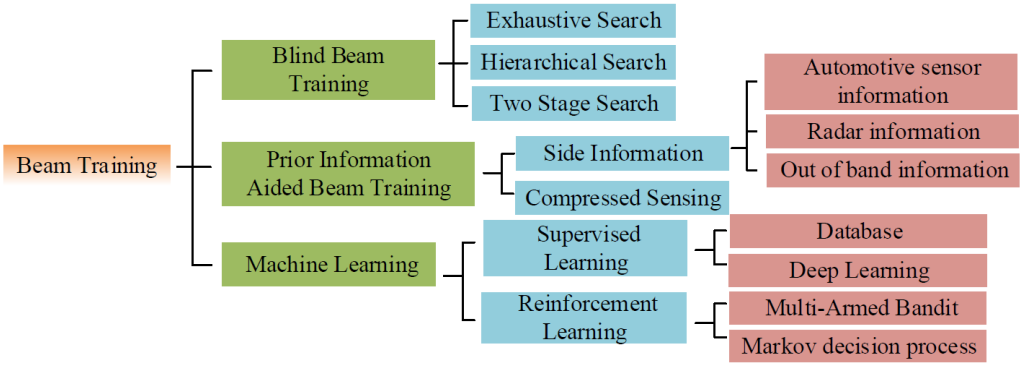


Figure 3.10: SotA Beam Training families of methods [1]

the single-beam training and transmission. This limitation is resolves in [48] where an adaptive compressed sensing algorithm is adopted on Saleh Valenzuela geometric channel model for full channel estimation, followed by hybrid precoding based on quantized beam steering directions to directly approximate the channel's dominant singular values, which aligns with the MAC-layer protocol IEEE 802.11 ay. On the other hand, in [41], authors investigate the performance of Regularized Zero Forcing for massive MIMO in addition to hybrid MMSE based precoding and combining techniques in [42]. Quite similar hybrid approach is found in [43], endorsed with one joint beam selection combining using Lens antenna array, attached to an adaptive Selection network, multiple RF chains and base band signal processing architecture. In [44], authors rely on 3D SVD based codebooks to align beams. All these methods require hybrid architectures, full channel estimation and a high overhead scaling with the MIMO massive configuration. These SVD based codebooks where compared to a deep neural network in [45], aiming to encounter the pilot overhead problem, still through hybrid architecture and the prior full-exchange of CSI. In [47], authors presented an overview of signal processing tools for mmWave BA, covering several channel/system models, the beam space sparsity in narrowband and wideband models, analog, digital and hybrid precoding and combining in addition to a deep dive through beam training protocols, including [2] [3], the low resolution receivers and the multi user extensions. In addition, in [49], the proposed spatially sparse precoding via orthogonal matching pursuit exploits the spatial structure of mmWave systems with large antenna arrays, formulating the problem as a sparse reconstruction problem, implemented on low-cost RF hardware with both limited and full feedback. Moreover, in [50], the dominant model in the literature, Saleh Valenzuela, is well investigated and the notions of its power delay profiles, fading and statistical properties are rigorously defined. The 802.11ay standard is investigated in [54] where authors merged adaptive compressed sensing solution with sub-connected hybrid precoding multi-resolution hierarchical codebook, for single path and multi path channels, while varying the power allocation, the level of interference and the quantization errors, comparing the performances with Exhaustive BA.

Therefore, a commonality observed in most conventional methods lies in their dependency on channel state information for channel estimation and Exhaustive

Beam Alignment techniques (using all available beam patterns). The impracticality of these approaches becomes evident in Massive MIMO setups, primarily due to the significant pilot overhead involved. In this context, Machine Learning based Beam Alignment emerges as a promising solution, offering potential avenues to overcome these inherent limitations. 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 among a partial point-to-point BA procedure, similar to the proposed approach in the first contributions of this PhD work. Besides, Deep learning through shallow neural networks is increasingly used by Wireless Communication scientists where we distinguish two learning paradigms: first, the ML methods related to the Supervised Learning via 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][15][16], auto-encoders for beam Management in [17] and several other neural architectures. Second, the Reinforcement Learning in [18][19][20], generally used to resolve the problems of Multi-Armed Bandit and Markov decision process. In [51], authors study systems having one dominant cluster and propose a deep neural network based on (AoD, AoA) beam patterns, using MMSE as QoS metric of evaluation, at high SNR regime. The limitations of this approach is lack of robustness handling noisy environments and lower SNR regimes. In [52], authors proposed a multi layer perceptron based algorithm for beam alignment in multi-path environments, using only phase-less received power measurements. Using 60 GHz radios with 36-element phased arrays, their algorithm suggests a 62% reduction in overhead, benchmarking the Exhaustive BA. Additionally, the survey in [53] provides a comprehensive overview of several emerging technologies for 5G and towards 6G systems, such as multiple access technologies, hybrid precoding and combining, non-orthogonal multiple access, cell-free massive MIMO, simultaneous wireless information and power transfer technologies, comparing existing wireless communication techniques like sub-6-GHz WiFi and sub-6 GHz 4G LTE over mmWave communications.

In essence, machine learning models offer a promising avenue for addressing the substantial pilot overhead by efficiently utilizing sub-sampled codebooks and a limited set of training data. However, their implementation often demands hybrid precoding architectures featuring multiple radio frequency chains, leading to moderate to high architectural complexity. Additionally, managing extensive datasets and the offline computational intensity associated with cross-validation pose significant challenges. Certain models advocated in existing literature, such as dense neural networks, add computational intricacy, prompting researchers to explore avenues for compression or delve into shallower architectures with reduced dimensions and parameters.

Benchmark 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 signal-

ing overhead scales as the product of the UE and BS codebook sizes. In the 60 GHz, the Exhaustive BA has been adopted in several mmWave $WLAN$ or $WPAN$ communication technologies, e.g., IEEE 802.15.3c [2], IEEE 802.11ad [3]. It is conventionally being applied in small MIMO configurations using small codebook sizes (e.g., codebooks of size 8×8 for LTE) and guarantees the optimal performance. For cellular networks [4], V2X communications, Unmanned Aerial Vehicle or High Speed Train applications, the infeasibility of brute force based BA pushes scientists to reduce the large signaling overhead, resulted from using massive antennas systems, aiming to find the smallest possible subset of beams that guarantees accurate Alignment and a reliable initial-link [1].

3.3.5 Beam Sweeping:

Beam Sweeping is a technique used in the initial phase of Beam Alignment to explore different beam directions and identify the optimal beamforming vector. It involves systematically steering a set of predefined beamforming vectors across the angular space to cover a wide range of potential beam directions. The receiver measures the received signal quality for each beam direction, such as the received signal strength or signal-to-noise ratio, and provides feedback to the transmitter. By sweeping through various beam directions, the system can identify the direction with the strongest signal and select the corresponding beamforming vector for subsequent data transmission.

Beam Sweeping is necessary in mmWave MIMO systems due to the narrow beam width and highly directional nature of mmWave signals. The main objective is to explore different beam directions and evaluate the quality of the received signals to identify the best beamforming direction for optimal communication. Beam Sweeping can be performed in a structured manner, such as using predefined codebooks with specific beamforming vectors, or in a more adaptive way, where the beam directions are dynamically adjusted based on the feedback from the receiver. The effectiveness of Beam Sweeping directly impacts the overall system performance, as it determines the accuracy and efficiency of subsequent Beam Alignment and data transmission processes.

3.3.6 Beam Tracking

Beam Tracking is a critical operation in mmWave MIMO communication systems that aims to maintain a robust and reliable communication link between the transmitter and receiver as they move relative to each other. Unlike Beam Alignment, which focuses on the initial establishment of an optimal beamforming link, Beam Tracking is responsible for continuously adapting the beamforming direction to compensate for the changing channel conditions caused by mobility, fading, and other environmental factors.

In Beam Tracking, the transmitter and receiver continuously exchange feedback information to estimate the variations in the channel and adjust the beamforming vectors accordingly. This feedback can include measurements of received signal

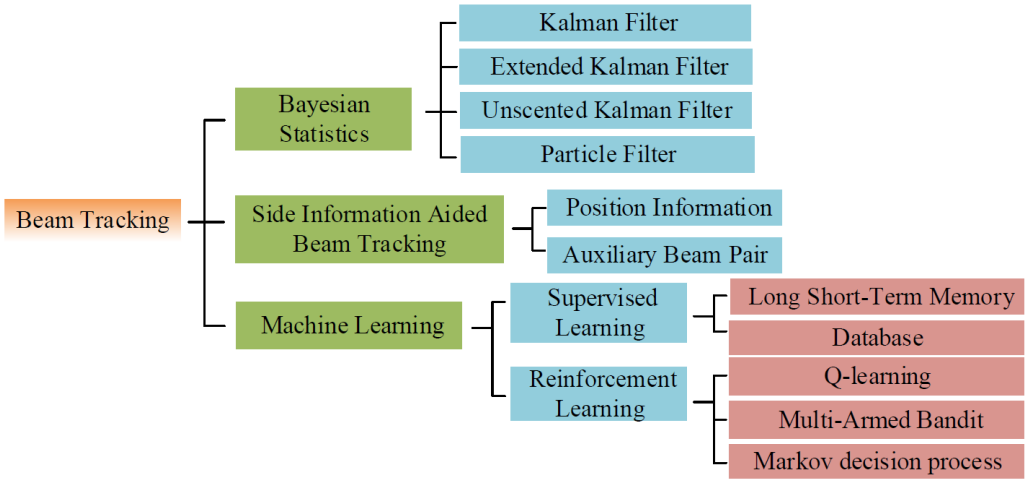


Figure 3.11: SotA Beam Tracking families of methods [1]

strength, signal quality indicators, or other channel parameters. Based on the feedback, the transmitter dynamically adjusts the beamforming direction to maintain an optimal link, ensuring high-quality communication and mitigating the effects of channel fluctuations. The families of state of the art methods used to encounter the Beam Tracking problem are resumed in figure (3.11):

The key challenge in Beam Tracking lies in accurately tracking the changing channel conditions in real-time and promptly adjusting the beamforming direction. This requires efficient feedback mechanisms, fast beamforming adaptation algorithms, and low-latency control signaling. Beam Tracking algorithms should be able to handle rapid variations in the channel state, such as fast-moving users or dynamic multi-path propagation, to ensure uninterrupted and reliable communication. By continuously tracking and adapting the beamforming direction, Beam Tracking enhances the system's robustness, improves link reliability, and helps maintain optimal performance in dynamic mmWave environments.

3.4 Machine Learning meets the beam Alignment Problem

ML techniques have emerged as a promising approach for addressing Beam Alignment problems in wireless communication systems. This trend is driven by the robustness and flexibility of ML approaches, which have shown superior performance compared to traditional methods and benchmarks in terms of Quality of Service [1]. By leveraging ML, researchers are able to provide concrete solutions to the challenges faced by conventional methods.

Although ML techniques may lack interpretability and pose challenges in hardware implementation, they remain a vital research direction for 5G and beyond. Particularly in the context of BA, ML tools offer a means to tackle the issue of large signaling overhead ratios that arise from using a significant number of pilots in high-

dimensional MIMO setups. Through various paradigms and learning approaches, ML algorithms efficiently capture the hidden interactions between transmitters and receivers and extract meaningful features to make accurate predictions.

Furthermore, ML's robustness has been empirically demonstrated in various experimentally challenging environments and scenarios, including Blind BA [55]. Unlike traditional methods that heavily rely on channel estimation prior to the alignment procedure, which can be time-consuming and resource-intensive in massive MIMO systems, ML approaches handle the alignment problem without any knowledge of the channel coefficients. The empirical evidence suggests that the impact of blindness in ML-based approaches is minimal, highlighting their effectiveness in addressing BA challenges.

3.4.1 Wireless communications datasets for AI tools

Data plays a pivotal role in the realm of Artificial Intelligence, particularly in the context of Data-driven Beam Alignment problems. These problems involve the utilization of datasets that can be categorized into two major groups. The first category comprises structured datasets generated based on well-established models found in the literature such as in survey [1] and in [56]. For example, in mmWave MIMO communications, widely used models like geometrical Saleh-Valenzuela [22] and its extensions, are employed. These datasets can take the form of matrices or tensors, containing various parameters such as Signal-to-Noise Ratio, Signal-to-Interference-plus-Noise Ratio, Received Signal Energy, beam patterns, GPS locations, spatial coordinates, and more. The values in these datasets are generated following rigorous model setups with multiple architectural parameters, simulating both indoor and outdoor scenarios [57].

The second category consists of datasets derived from experimental measurements in real-life scenarios. Notably, datasets like deepMIMO [58], QuadriGa [59], DeepSense [60] have gained significant recognition in the field, providing a rich collection of technologies, scenarios, and source codes. These datasets also encompass a wide range of indoor and outdoor simulations and incorporate diverse types of data, including images, videos, radar/Lidar signals, sensor readings, and others [61]. Typically, the first category of datasets is utilized in the initial stages of AI model validation. Once the researcher verifies the theoretical functionality of their model, transitioning to datasets akin to deepMIMO is recommended for enhanced robustness and more rigorous evaluation. However, scientist can face a lack of accessibility to industrial datasets, protected by privacy and confidentiality measures. Some of these datasets are generally shared as open-source for academic research and scientific explorations. Huge datasets require a lot of training time for ML models, even the low complex ones where the debate of Offline and Online training and fine-tuning is continuously investigated for the industrial deployment of AI tools in Wireless Communications components.

In this PhD manuscript, all proposed mmWave MIMO data-driven BA models are based on the Saleh-Valenzuela framework, which is widely used in the literature for both classic and ML based BA. Depending on the nature of the data, whether

it is continuous or discrete, a distinction is made between non-linear regression problems and logistic regression. These models and datasets serve as foundations for developing AI-driven BA techniques, allowing for accurate and efficient beam alignment in mmWave MIMO systems. The next sections introduce the two families of optimization problems, illustrating the foundations of the proposed ML tools in the literature: non-linear regression problems and logistic-regression problems.

3.5 Non-linear regression using shallow neural networks

Regression problems in the context of machine learning refer to the task of predicting a continuous output value based on input features. These problems involve establishing a functional relationship between the input variables and the target variable, allowing for the estimation of unknown values or making future predictions. The goal of regression analysis is to identify and quantify the relationships between the input variables and the target variable, enabling the creation of a predictive model that can generalize to unseen data. Various algorithms and techniques are employed in regression, such as linear regression, polynomial regression, and support vector regression, each suited to different types of data and underlying assumptions. Non-linear regression and MSE loss are two fundamental keywords of this manuscript:

- Non-linear regression expands upon the concept of regression by considering complex relationships between the input variables and the target variable. Unlike linear regression, which assumes a linear relationship, non-linear regression models allow for more flexible and intricate mappings. These models employ non-linear functions to capture the underlying patterns and dependencies in the data. Non-linear regression is particularly useful when the relationship between the variables cannot be adequately represented by a linear equation. It enables the modeling of curvilinear, exponential, logarithmic, or other non-linear trends, offering a more accurate representation of complex real-world phenomena.

- MSE loss function, mathematically introduced in the previous chapter, is particularly suitable for non linear regression problems as long as it aims to minimize the discrepancy between the predicted values and the ground truth by penalizing large deviations. It provides a quantitative measure of the model's performance, with lower MSE values indicating a better fit to the data. Thus, it focuses on the magnitude of errors and assigns higher weights and learning parameters to larger errors due to the squaring operation. The optimization process in regression models often involves minimizing the MSE loss through techniques like gradient descent, enabling the model to converge towards optimal parameter values. The MSE loss function is widely used due to its mathematical properties, interpretability, and compatibility with various optimization algorithms.

In the context of neural networks and deep learning, the mean squared error loss serves as a measure of the discrepancy between the network's predicted output and the true target value. By calculating the average squared difference across all training samples, the MSE loss guides the network's learning process, facilitating the adjustment of the network's weights and biases to minimize the overall error. The MSE loss function provides a continuous and differentiable objective that enables efficient optimization through backpropagation and gradient-based algorithms. Its utilization in neural network training ensures the network's ability to learn and generalize from the training data, making it a fundamental component in regression tasks within the deep learning domain.

3.6 Logistic regression using classifiers

Logistic regression is a statistical modeling technique used for binary classification problems. It is primarily employed when the target variable is categorical and has two possible outcomes, such as "yes" or "no," "true" or "false," or "spam" or "not spam." The goal of logistic regression is to estimate the probability of a given input belonging to a specific class. Unlike linear regression, which predicts continuous values, logistic regression utilizes a logistic function (also known as the sigmoid function) to map the linear combination of input features to a value between 0 and 1, representing the probability of belonging to the positive class. The logistic regression model is trained using maximum likelihood estimation, optimizing the parameters to maximize the likelihood of the observed data. It is a popular and interpretable algorithm in the field of machine learning, often used as a baseline for more complex classification models.

- Logistic regression is commonly used for binary classification problems, where the target variable has two possible outcomes. However, it can also be extended to handle multi-class classification tasks through various techniques, such as one-vs-rest or softmax regression. In one-vs-rest (or one-vs-all) logistic regression, a separate logistic regression model is trained for each class, treating it as the positive class and the remaining classes as the negative class. During prediction, the model that produces the highest probability is selected as the predicted class. Moreover, Softmax regression, also known as multinomial logistic regression, is another extension of logistic regression for multi-class classification. Instead of training separate models for each class, softmax regression uses a single model with multiple output nodes, each corresponding to a class. The softmax function is applied to the output layer, which converts the raw predictions into a probability distribution across all classes. The class with the highest probability is then assigned as the predicted class.

- Classification, on the other hand, is a fundamental task in machine learning that involves assigning input samples to predefined categories or classes. It is the process of learning a mapping from input features to discrete output labels. The objective of a classification model is to accurately classify unseen

data based on the patterns and relationships learned from the training data. Various algorithms are employed for classification, such as logistic regression, decision trees, support vector machines, random forests, and neural networks. The performance of classification models is typically evaluated using metrics such as accuracy, precision, recall, and F1-score, which quantify the model's ability to correctly classify samples into their respective classes.

The need for classification problem formulation came in when we considered Quantizing the dataset before the Alignment procedure. Therefore, the data becomes discrete. We then use cross-entropy as our loss function which measures the dissimilarity between the predicted probability distribution and the true probability distribution of the target classes. The cross entropy loss is derived from information theory and aims to minimize the average number of bits needed to encode the true class labels given the predicted probabilities. Mathematically, the cross entropy loss is computed by taking the negative sum of the logarithm of the predicted probabilities of the correct classes. This loss function encourages the model to assign high probabilities to the correct classes and penalizes deviations from the true distribution.

3.7 Conclusion

This chapter serves as a comprehensive introduction to the realm of Beam Alignment and Management in mmWave MIMO systems. We begin by providing an overview of the mmWave frequency band, highlighting its advantages and limitations in wireless communication. Next, we delve into the essential concepts and keywords associated with the signal processing package of Beam Management techniques. We also explore the interdisciplinary intersection between Machine Learning and Beam Alignment, discussing the nature of datasets and the choice of loss function, which in turn determines the regression problem to be addressed. In these two introductory chapters on ML from one side and BA on the other hand, we have provided a comprehensive description of all the tools required for a full understanding of the manuscript. In the subsequent chapter, we present the first contribution of this thesis, where we apply ones of these tools, Matrix Factorization and its variants, in a point-to-point narrowband system model. Our objective is to tackle the challenge of large signaling overhead encountered in the exhaustive Beam Alignment process.

Chapter 4

Matrix Factorization for blind and partial Beam Alignment in massive mmWave MIMO

"Mathematics compares the most diverse phenomena and discovers the secret analogies that unite them."

Joseph Fourier

4.1 Introduction

This chapter introduces the first scientific contribution of the thesis, which centers around addressing the substantial signaling overhead challenge through two key words: "Partial" Beam Alignment and "Blind" Beam Alignment.

- Partial Beam Alignment, the first keyword of this chapter, derives its name from its distinctive approach of not utilizing the entire set of beam pairs for alignment, a departure from conventional methods. Instead, we leverage codebooks that store beam patterns (AoA, AoD) between the User Equipment and Base Station. These codebooks are judiciously sub-sampled, and a small training set is randomly selected, determined by a predefined ratio known as the overhead ratio. The optimal value of this ratio represents the initial research direction of this chapter.
- Blind Beam Alignment, our second focal point, revolves around sounding beam pairs between the UE and BS without necessitating the exchange of Channel State Information. This approach renders the system "blind" to channel details.

We propose to track the training and test performance of the first proposed learning tool, MF, over a basic point-to-point narrowband LoS channel and a simple fully-analog architecture with one RF chain at both sides of the transmission with

the perspective of extending the channel and system model progressively throughout the thesis in a parallel way to investigating more ML models. Thus, the proposed Beam Alignment methodology in this chapter hinges on completing a sparse and low-rank Received Signal Energy matrix using Matrix Factorization, backed by robust theoretical convergence proofs. This approach not only optimizes signaling overhead but also lays the foundation for efficient and effective beam alignment in massive MIMO configurations, as evidenced by promising results both in our work and in related literature. This chapter comprises several key components, including the system architecture, the mathematical formulation of the partial and blind BA problem, the equations detailing the Matrix Factorization based solution, as well as experimental simulations and comprehensive performance evaluations.

4.2 Point-to-point system architecture with one-RF chain at UE and BS

We consider a mmWave Massive MIMO setup, where a UE and BS equipped with N_T and N_R antennas respectively, wish to align the optimal Tx-Rx beamformer and combiner pair in order to establish an initial link.

4.2.1 Beam former and combiner

The assumption of low-energy/complexity architecture is achieved, by having one RF chain at the UE and one RF chains at BS. Note that the number of RF chains should be smaller than the corresponding number of antenna and is fixed as one in this narrowband model regarding this chapter and is generalized to multiple chains within the wideband model of the next chapter. The UE selects its analog beamformer $\boldsymbol{f}_u \in \mathbf{C}^{N_T}$ from a codebook of (AoA, AoD) beam patterns, $u \in \mathcal{T}$, where the set $\mathcal{T}$, represents the UE codebook. Equivalently, the BS chooses its fully analog combiner $\mathbf{w}_i \in \mathbf{C}^{N_R}$ from a codebook $i \in \mathcal{R}$, where $\mathcal{R}$ represents the set of BS codebook indexes. We denote by $C_T = |\mathcal{T}|$, and $C_R = |\mathcal{R}|$ the cardinality of codebook at the UE and BS, resp. Indeed, UE and BS beams satisfy the constant modulus constraints:

$$\boldsymbol{f}_u \in \mathbf{C}^{N_T} \ , \ | \ [\ \boldsymbol{f}_u \ ]_t \ | = (N_T)^{-1} \ , \ \forall \ t \in \{1, ..., N_T\}$$

$$\boldsymbol{w}_i \in \mathbf{C}^{N_R} \ , \ | \ [\ \boldsymbol{w}_i \ ]_r \ | = (N_R)^{-1} \ , \ \forall \ r \in \{1, ..., N_R\}$$

where $[\ \mathbf{x} \ ]_t$ is entry t of a vector $\mathbf{x}$ and $|x|$ is the absolute value of x.

4.2.2 Narrowband Saleh-Valenzuela mmWave Channel model

We define *beam-pair* $(u, i) \in \mathcal{T} \times \mathcal{R}$, as beam $u \in \mathcal{T}$ from the UE codebook, and beam $i \in \mathcal{R}$ from the BS codebook. In addition, as in [62] the signal at the BS corresponding to beam couple (u, i) is expressed as:

$$y_{u,i} = \mathbf{w}_i^H \mathbf{H} \boldsymbol{f}_u s_u + \tilde{n}_i, \ \forall \ (u, i) \in \mathcal{T} \times \mathcal{R}, \tag{4.1}$$

where $s_u = \sqrt{P_u}$ is the pilot symbol associated with $\boldsymbol{f}_u$, P_u is the corresponding transmitted power over $\tilde{n}_i = \mathbf{w}_i^H \mathbf{n}$, the zero mean additive white Gaussian noise with unit variance, $\tilde{\sigma_i}^2 = 1$. We denote the MIMO channel by $\mathbf{H} \in \mathbf{C}^{N_R \times N_T}$, that may be a narrow- or wide-band channel model in general, narrowband in the numerical simulations of this chapter, and $L := \text{rank}(\mathbf{H})$ its rank (where $L \ll (N_T, N_R)$). Moreover, the channel is invariant for a mmWave coherence interval, I. It is called in the literature slow fading channel [22] and is expressed thanks to the following equation [63]:

$$\mathbf{H} = \sqrt{\frac{N_T N_R}{L}} \sum_{i=1}^{L} \rho_i \ \mathbf{a}_R(\theta_i^{(R)}) \mathbf{a}_T^H(\theta_i^{(T)}) \tag{4.2}$$

where L is number of paths of the channel representing the channel Rank, $\theta_i^{(R)}$ and $\theta_i^{(T)}$ are angles of arrival at the BS, and angles of departure at the UE (AoA/AoD) of the i^{th} path, respectively. They are both considered uniform over $[-\pi/2,\ \pi/2])$. Moreover, ρ_i is the complex gain of the i^{th} path such that $\rho_i \sim \mathcal{CN}(0,1),\ \forall i$. Finally, $\mathbf{a}_R(\theta_i^{(R)}) \in \mathbf{C}^{N_R}$ and $\mathbf{a}_T(\theta_i^{(T)}) \in \mathbf{C}^{N_T}$ are the array response vectors at both the UE and BS, respectively. We further assume that $\mathbf{H}$ is static during the BA procedure, and that $\mathbf{H}$ is completely unknown to UE and BS where an independent realization is observed for each time-index i.e., $\mathbf{H}^{(1)} \approx ... \approx \mathbf{H}^{(I)}$.

4.2.3 Received Signal Energies

We define the received SNR for the beam couple (u, i) as:

$$\text{SNR}_{u,i} = P|\mathbf{w}_i^H \mathbf{H} \boldsymbol{f}_u|^2,\ \forall\ (u,i) \in \mathcal{T} \times \mathcal{R} \tag{4.3}$$

We drop the time index from $\mathbf{w}_i, \mathbf{f}_u, P_u$ since it is implicitly present in these quantities. However, due to the lack of CSI due to the choice of the blind approach, the BS is unable to compute the exact value of SNR. Thus, the BA is based on approximating these SNR values with received signal energies or received signal strengths as in [63]:

$$RSE_{u,i} = |y_{u,i}|^2,\ \forall (u,i) \in \mathcal{T} \times \mathbf{R}, \tag{4.4}$$

which does not require knowing $\mathbf{H}, \mathbf{f}_u, P_u$. Therefore, we implicitly assume that the RSE is close to the true SNR, for each beam-pair, i.e., $\text{SNR}_{u,i} \approx \text{RSE}_{u,i}, \forall (u,i) \in \mathcal{T} \times \mathcal{R}$ just like a majority of blind BA approaches presented in the literature. Note that the impact of this approximation is trivial when we use ML, where learning models have the ability to extract hidden information from the RSE distribution and provide performances quasi-similar to the ones where the exact values of SNR are used.

4.2.4 Benchmark

Exhaustive BA: is a SotA method against which we benchmark, throughout the whole PhD manuscript. Recall that Brute Force BA is performed via *exhaustively*

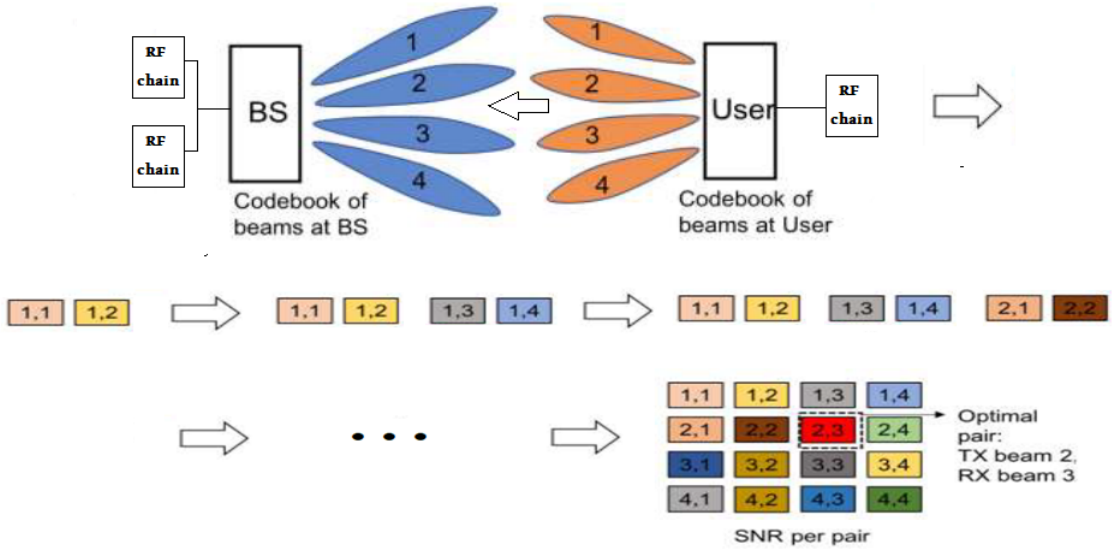


Figure 4.1: Exhaustive Search step by step using two RF chains at BS and one RF chain at UE through a 4×4 MIMO setup

sounding both codebooks at UE and BS, i.e., exhaustively testing each and every beam-pair, jointly as shown in figure (4.1) [2] [3]. Then, we perform an exhaustive search of the beam-pair that holds the highest RSE.
The indexes of the optimal beam-pair are selected as:

$$(u^\star, i^\star) = \underset{(u,i)\in\mathcal{T}\times\mathcal{R}}{argmax}\ RSE_{u,i} \tag{4.5}$$

From the above equation, it is clear that exhaustive BA must sound each and every beam-pair in the UE and BS codebooks, $\mathcal{T}\times\mathcal{R}$: consequently, its *signaling overhead* is, $|\mathcal{T}| \times |\mathcal{R}| = C_T\ C_R$. Therefore, the main challenge of exhaustive BA is the massive signaling overhead, that scales as the product of the codebook sizes at UE and BS.

4.3 Problem Statement

Instead of processing exhaustive alignment for all possible beam pairs at UE and BS, the proposed approach offers the capability to selectively probe a subset of these beams while accurately predicting the RSE for the unsounded beam pairs. This approach stands in contrast to several existing methods that rely on channel knowledge. Furthermore, we achieve this with just a single Radio Frequency chain to control the extensive antenna arrays at both the transmitter and receiver, resulting in minimal power consumption.

Moreover, our approach avoids the use of digital precoding or combining and instead leverages full analog beamforming. As a result, our method and derivations are equally applicable to the wideband channel model. Notably, we aim to sound a very small subset of the codebooks, resulting in a significant reduction in signaling

overhead compared to the exhaustive BA benchmark, while maintaining negligible loss in optimality.

As mentioned in the system model, we capitalize on the low-rank structure of the Received Signal Energy matrix, denoted as $\mathbf{S}$. This low-rank structure arises from the inherent low-rank nature of the channel matrix $\mathbf{H}$. Specifically, we suggest sounding a subset of beam pairs and using their corresponding RSE values as a training set to optimize a low-rank MF model. This trained model can then predict the RSE values for the beam pairs that were not sounded. Note that the minimum number of training samples required is always proportional to the rank and dimensions of $\mathbf{S}$. This property makes low-rank MF ideally suited for selecting a small subset of beam pairs and predicting the RSE for the larger number of remaining unknown beam pairs.

4.3.1 Proposed low-rank MF Approach:

Instead of sounding the entire codebooks at UE and BS, $\mathcal{T}$ and $\mathcal{R}$, e.g., Exhaustive BA described above, we use *sub-sampled codebooks* of beam couple, $\mathcal{R}_\mathcal{S}$ and $\mathcal{T}_\mathcal{S}$ such that, $\mathcal{R}_\mathcal{S} \subset \mathcal{R}$ and $\mathcal{T}_\mathcal{S} \subset \mathcal{T}$. The sub-sampled codebooks are chosen to have small sizes, i.e., $|\mathcal{R}_\mathcal{S}| \ll |\mathcal{R}|$ and $|\mathcal{T}_\mathcal{S}| \ll |\mathcal{T}|$. We denote *beam-pair* (u, i) as the combination of analog beamforming vector i in the UE codebook of beams, and analog combining vector u in the BS codebook of beams. Afterwards, we only sound beam-pairs from sub-sampled codebook of beams, $\mathcal{R}_\mathcal{S}$ and $\mathcal{T}_\mathcal{S}$ (see Fig. 4.2). Therefore, we express the RSE of beam-pair (u, i), from the sub-sampled codebooks $\mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S}$, as:

$$RSE_{u,i} := \mid y_{u,i} \mid_2^2 \forall\ (u, i) \in \mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S} \tag{4.6}$$

where $y_{u,i}$ is the received signal at BS in an Uplink scenario, resulting from beam-pair (u, i), given in (4.1) . (6.2) can be equivalently written using an incomplete RSE matrix, $\mathbf{S} \in \mathbb{R}^{C_T \times C_R} (:= \mathbb{R}^{|\mathcal{T}| \times |\mathcal{R}|})$ as,

$$[\ \mathbf{S}\ ]_{u,i} := \begin{cases} \text{RSE}_{u,i} & , \text{ if } (u, i) \in \mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S} \\ \text{unknown} & , \text{ if } (u, i) \notin \mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S} \end{cases} \tag{4.7}$$

where $[\mathbf{S}]_{u,i},\ \forall (u, i) \in \mathcal{T} \times \mathcal{R}$ denotes element (u, i) of $\mathbf{S}$.

Indeed, the BS knows the RSE of entries in $\mathbf{S}$ that correspond to the sub-sampled codebooks at UE and BS, $\mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S}$, because they already have been sounded. Subsequently, The value of RSE is undefined for the beam-pairs that were not sounded and shows the unknown matrix entries in the Matrix Completion task. We will use their RSE as the *training set*, $\mathcal{K}$, which we introduce as the sub-sampled codebook indexes, $\mathcal{K} := \{(u, i) \mid (u, i) \in \mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S}\}$. We then use the training set, $\mathcal{K}$ (beam-pairs that have sounded and known RSE values in the matrix $\mathbf{S}$, in (4.7)) to learn a low-rank MF model, and apply it, to predict the RSE of entries in $\mathbf{S}$ that are labeled 'unknown', in (4.7); see Fig 4.2. In the matrix of RSE, $\mathbf{S}$, we rely on the entries for which the RSE is sounded and known (i.e., the training set $\mathcal{K}$) to predict the RSE value of unknown beam-pairs.

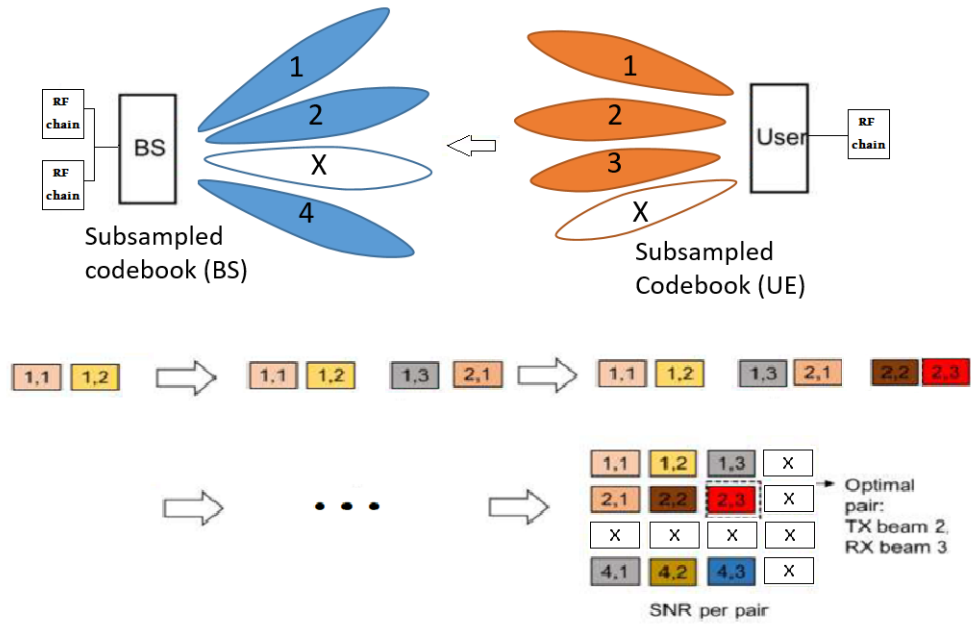


Figure 4.2: Proposed partial BA using sub-sampled codebooks: toy-example with $C_T = C_R = 4$ using one RF chain at UE and two RF chains at BS
i) Randomly sound subset of beam-pairs from codebook at UE and BS (colored entries in the dataset matrix represent the training set)
ii) Process MF to predict RSE of non-sounded beam-pairs (matrix coefficients marked with X)
iii) Select the optimal couple which holds the largest RSE (or SNR in case of prior CSI-based channel estimation)

The intuition behind low-rank MF is to model the RSE of the sounded beam-pairs, i.e., the coefficients of $\mathbf{S}$ that are known $\mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}$, as an inner product between two D-dimensional latent factors, i.e. latent vectors, $\boldsymbol{\theta_u}, \boldsymbol{\psi_i}$. Thus, the RSE of beam-pair (u, i) is formulated as:

$$\begin{aligned} RSE_{u,i} = [\mathbf{S}]_{u,i} := \boldsymbol{\theta_u}^T \boldsymbol{\psi_i} \ , \boldsymbol{\theta_u} \in \mathbb{R}^D \ , \ \boldsymbol{\psi_i} \ \in \ \mathbb{R}^D, \\ \forall \ (u,i) \in \mathcal{K} (:= \mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}) \end{aligned} \tag{4.8}$$

where D is the dimension of the MF model (also referred to as the complexity of the learning model). Due to the low-rank MF model, D is theoretically assumed to much smaller than the dimensions of $\mathbf{S}$, i.e., $D \ll (C_T, C_R)$. The goal of low-rank MF method is to optimize the latent factors, $\{\boldsymbol{\theta_u}, \boldsymbol{\psi_i}\}_{(u,i)\in\mathcal{K}}$, corresponding to the coefficients of $\mathbf{S}$ that are known (i.e., the training set $\mathcal{K} := \mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}$). Particularly, learning the MF latent factors, $\boldsymbol{\theta_u}, \boldsymbol{\psi_i}$, corresponding to sample (u, i), of the training set $\mathcal{K}$, is formulated as:

$$f_{u,i}(\boldsymbol{\theta_u}, \boldsymbol{\psi_i}) := \left(\ [\mathbf{S}]_{u,i} - \boldsymbol{\theta_u}^T \boldsymbol{\psi_i} \ \right)^2 = \left(\ SNR_{u,i} - \boldsymbol{\theta_u}^T \boldsymbol{\psi_i} \ \right)^2 \tag{4.9}$$

where $f_{u,i}(\boldsymbol{\theta_u}, \boldsymbol{\psi_i})$ is the cost function of the beam-sample (u, i) of the training set, $\mathcal{K}$. Afterwards, we sum over all the samples (u, i) of the training set $\mathcal{K}$, to get the total cost function, and formulate the optimization problem (P1) that results from the low-rank MF problem, as:

$$(P1) := \begin{cases} \underset{\{\boldsymbol{\theta_u}, \boldsymbol{\psi_i}\}_{(u,i)\in\mathcal{K}}}{\text{argmin}} \ \sum_{(u,i)\in\mathcal{K}} f_{u,i}(\boldsymbol{\theta_u}, \boldsymbol{\psi_i}) \\ \text{subject to } \boldsymbol{\theta_u} \in \mathbb{R}^D, \ \boldsymbol{\psi_i} \in \mathbb{R}^D \end{cases}$$

where the loss function of $(P1)$ is expressed as:

$$\sum_{(u,i)\in\mathcal{K}} f_{u,i}(\boldsymbol{\theta_u}, \boldsymbol{\psi_i}) = \sum_{(u,i)\in\mathcal{K}} \left(\ [\mathbf{S}]_{u,i} - \boldsymbol{\theta_u}^T \boldsymbol{\psi_i} \ \right)^2 \tag{4.10}$$

Note that in $(P1)$, the RSE of the sounded beam-pairs are known (training set, $\{[\mathbf{S}]_{u,i} | \forall (u,i) \in \mathcal{K}\}$), and the optimization variables that need to be tuned are the latent factors w.r.t the training set, $\{\boldsymbol{\psi_i}, \boldsymbol{\theta_u} | \forall (u,i) \in \mathcal{K}\}$.

4.3.2 Proposed low-rank NMF Approach:

Our proposed Non-Negative Matrix Factorization follows similar steps as MF, with the main difference of constraining the latent factors of the NMF model to be non-negative, i.e., the RSE of beam-pair $(u, i) \in \mathcal{K}$, assuming the NMF model, is given as:

$$\begin{aligned} RSE_{u,i} = [\mathbf{S}]_{u,i} := \boldsymbol{\theta_u}^T \boldsymbol{\psi_i} \ , \boldsymbol{\theta_u} \in \mathbb{R}_+^D \ , \ \boldsymbol{\psi_i} \in \mathbb{R}_+^D, \\ \forall \ (u,i) \in \mathcal{K} (:= \mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}) \end{aligned} \tag{4.11}$$

The logic leading from the above RSE based model to the NMF optimization problem, are the same those of MF. We skip re-writing them in this section to

avoid redundancy. Thus, we formulate $(P2)$, the NMF optimization problem, as follows:

$$(P2) := \begin{cases} \underset{\{\boldsymbol{\theta_u}, \boldsymbol{\psi_i}\}_{(u,i)\in\mathcal{K}}}{\text{argmin}} & \sum_{(u,i)\in\mathcal{K}} f_{u,i}(\boldsymbol{\theta_u}, \boldsymbol{\psi_i}) \\ \text{subject to } \boldsymbol{\theta_u} \geq \mathbf{0},\ \boldsymbol{\psi_i} \geq \mathbf{0} \end{cases}$$

where $\mathbf{0}$ is the all-zero vector of dimension D. Note that the loss function of $(P2)$ is similar as that of $(P1)$, given in (4.10).

4.3.3 Overhead ratio

The overhead ratio is the most important parameters of this chapter and probably the whole PhD manuscript. It is expressed as:

$$\eta := \frac{\text{overhead of learning based BA}}{\text{overhead of exhautive BA}} = \frac{|\mathcal{T}_\mathcal{S}| \times |\mathcal{R}_\mathcal{S}|}{|\mathcal{T}| \times |\mathcal{R}|} \tag{4.12}$$

It measures the signaling overhead of all the proposed MF/NMF methods compared to that of our benchmark, the exhaustive BA (where $0 < \eta \leq 1$). Evidently, we desire the smallest possible value of η, resulting consequently in the smallest signaling overhead, which is vital for massive MIMO configurations. On the other hand, low η indicates that the dimensions of the training set is small and consequently, the prediction error may be large. For that reason, there is a major trade-off between η value and the prediction error. Note that the signaling overhead ratio from an ML perspective is the train/test split of our dataset. The RSE matrix $\mathbf{S}$ includes the mmWave MIMO channel, $\mathbf{H}$. Thus, $\mathbf{S}$ is a large-dimensions matrix $((C_T, C_R) \gg 1)$, characterized with low-rank structure $(\text{rank}(\mathbf{S}) \ll (C_T, C_T))$. Thus, low-rank MF/NMF both fit to the task in question with theoritical guarantees of the monotonic convergence of the loss function to minimize.

4.4 Solutions to the formulated problems

We resolve the MF problem $(P1)$ using the following methods:

- Block Coordinate Descent, i.e., Alternating Least Squares.
- BCD with Stochastic Gradient Descent.
- Block Gradient Descent merging Gradient Descent and BCD definitions.

Besides, figure (4.3) provides a simplified diagram representation so that we highlight the learning and completion procedure: MF learns from known coefficients in order to predict for the unknowns with the fundamental condition of low-rank input dataset matrix.

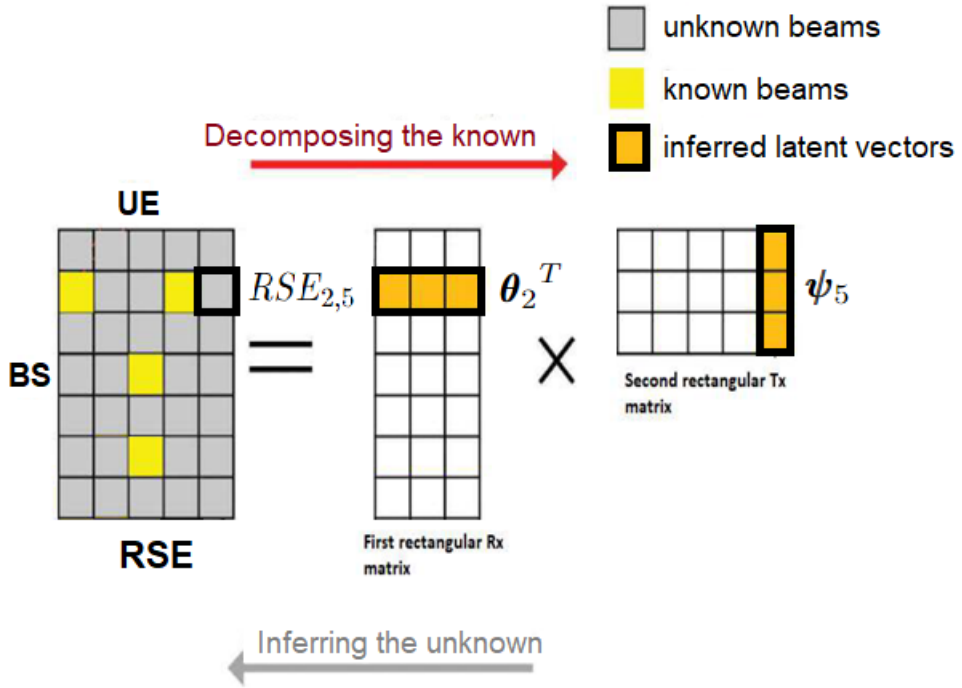


Figure 4.3: Toy Example: Matrix Factorization with $|\mathcal{T}| = 5, |\mathcal{R}| = 7, D = 3$. MF results in two rectangular matrices to be optimized: MF uses the RSE of known beams in yellow to complete and infer for the unknown beams, colored in gray. The product of the latent vectors $\boldsymbol{\theta}_2^T$ and $\boldsymbol{\psi}_5$ gives the unknown value of $RSE_{2,5}$

4.4.1 BCD, BGD and BSGD solutions using Matrix Factorization

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, that are solved iteratively. At iteration k, the sub-problem $(S1)$ is defined by fixing block $\{\boldsymbol{\psi_i}^{(k)}\}_{\forall i}$, and only solving block $\{\boldsymbol{\theta_u}\}_{\forall u}$:

$$(S1): \ \boldsymbol{\theta_u}^{(k+1)} = argmin_{\boldsymbol{\theta_u} \in \mathbb{R}^d} \ f(\{\boldsymbol{\theta_u}, \boldsymbol{\psi_i}^{(k)}\}) \tag{4.13}$$

$$= \sum_{(u,i)\in\mathcal{K}} [([\mathbf{S}]_{u,i} - \boldsymbol{\theta_u}^T \boldsymbol{\psi_i}^{(k)})^2 + \mu_u \|\boldsymbol{\theta_u}\|_2^2 + \lambda_i \|\boldsymbol{\psi_i}^{(k)}\|_2^2] \tag{4.14}$$

In addition, the sub-problem $(S2)$ is defined by fixing block $\{\boldsymbol{\theta_u}^{(k+1)}\}_{\forall u}$ in (P_1), and only updating block $\{\boldsymbol{\psi_i}\}_{\forall i}$:

$$(S2): \ \boldsymbol{\psi_i}^{(k+1)} = argmin_{\boldsymbol{\psi_i} \in \mathbb{R}^d} \ f(\{\boldsymbol{\theta_u}^{(k+1)}, \boldsymbol{\psi_i}\}) \tag{4.15}$$

$$= \sum_{(u,i)\in\mathcal{K}} [([\mathbf{S}]_{u,i} - \boldsymbol{\theta_u}^{(k+1)} \boldsymbol{\psi_i})^2 + \mu_u \|\boldsymbol{\theta_u}^{(k+1)}\|_2^2 + \lambda_i \|\boldsymbol{\psi_i}\|_2^2] \tag{4.16}$$

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

$$(S1) \ := argmin_{\boldsymbol{\theta_u} \in \mathbb{R}^d} \sum_{(u,i) \in \mathcal{K}} [[\mathbf{S}]^2_{u,i} - 2[\mathbf{S}]_{u,i} \boldsymbol{\theta_u}^T \boldsymbol{\psi_i}^{(k)} + \boldsymbol{\theta_u}^T \boldsymbol{\psi_i}^{(k)} \boldsymbol{\psi_i}^{(k)^T} \boldsymbol{\theta_u} + \mu_u \|\boldsymbol{\theta_u}\|_2^2] \tag{4.17}$$

$$\Leftrightarrow argmin_{\boldsymbol{\theta_u} \in \mathbb{R}^d} \sum_u [-2\boldsymbol{\theta_u}^T \sum_i ([\mathbf{S}]_{u,i} \boldsymbol{\psi_i}^{(k)}) + \boldsymbol{\theta_u}^T \sum_i (\boldsymbol{\psi_i}^{(k)} \boldsymbol{\psi_i}^{(k)^T}) \boldsymbol{\theta_u} + \mu_u \|\boldsymbol{\theta_u}\|_2^2] \tag{4.18}$$

$$\Leftrightarrow argmin_{\boldsymbol{\theta_u} \in \mathbb{R}^d} \sum_{u \in \mathcal{U}_i} [-2\boldsymbol{\theta_u}^T (\mathbf{r}_u^{(k)}) + \boldsymbol{\theta_u}^T (\mathbf{Q_u}^{(k)}) \boldsymbol{\theta_u} + \mu_u \|\boldsymbol{\theta_u}\|_2^2] = \sum_{u \in \mathcal{U}_)} h_u(\boldsymbol{\theta_u}), (d.1) \tag{4.19}$$

$$\boldsymbol{\theta_u}^{(k+1)} = argmin_{\boldsymbol{\theta_u} \in \mathbb{R}^d} \ [-2\boldsymbol{\theta_u}^T \mathbf{r_u}^{(k)} + \boldsymbol{\theta_u}^T (\ \mathbf{Q_u}^{(k)} + \mu_u \mathbf{I_D}\ ) \boldsymbol{\theta_u}] \ = \ f_1(\boldsymbol{\theta_u}), (e.1) \tag{4.20}$$

$\forall u \in \mathcal{U}_i$ where $\mathcal{U}_i$ is the set of row-indexes u in the RSE matrix corresponding to column i in the known entries of 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 sub-problem S1, by finding the global min of $f_1(\boldsymbol{\theta_u})$:

$$\nabla f_1(\boldsymbol{\theta_u}) = \mathbf{0} \Leftrightarrow -2\mathbf{r_u}^{(k)} + 2(\ \mathbf{Q_u}^{(k)} + \mu_u \mathbf{I_D}\ ) \boldsymbol{\theta_u} = \mathbf{0} \Leftrightarrow \boldsymbol{\theta_u} = (\ \mathbf{Q_u}^{(k)} + \mu_u \mathbf{I_D}\ )^{-1} \mathbf{r_u}^{(k)} \tag{4.21}$$

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

$$(S2) : \boldsymbol{\psi_i}^{(k+1)} = argmin_{\boldsymbol{\psi_i} \in \mathbb{R}^d} \ [-2\mathbf{t_i}^{(k+1)^T} \boldsymbol{\psi_i} + \boldsymbol{\psi_i}^T (\ \mathbf{P_i}^{(k+1)} + \lambda_i \mathbf{I}\ ) \boldsymbol{\psi_i}] = f_2(\boldsymbol{\psi_i}), (e.2) \tag{4.22}$$

$\forall\ i \in \mathcal{I}_u$ where $\mathcal{I}_u$ is the set of column-indexes i in the RSE matrix corresponding to row u in the known entries of RSE matrix, $\mathbf{t_i}^{(k+1)} = \sum_u (\ [\mathbf{S}]_{u,i} \boldsymbol{\theta_u}^{(k+1)^T})$ and $\mathbf{P_i}^{(k+1)} = \sum_u (\ \boldsymbol{\theta_u}^{(k+1)} \boldsymbol{\theta_u}^{(k+1)^T}\ )$.

Afterwards, we derive a closed form solution for sub-problem S2, by finding the global min of $f_2(\boldsymbol{\psi}_i)$:

$$\nabla f_2(\boldsymbol{\psi_i}) = \mathbf{0} \Leftrightarrow -2\mathbf{t_i}^{(k+1)} + 2(\ \mathbf{P_i}^{(k+1)} + \lambda_i \mathbf{I_D}\ ) \boldsymbol{\psi_i} = \mathbf{0} \tag{4.23}$$

$$\Leftrightarrow \boldsymbol{\psi_i} = (\ \mathbf{P_i}^{(k+1)} + \lambda_i \mathbf{I_D}\ )^{-1} \mathbf{t_i}^{(k+1)} \tag{4.24}$$

$$\leftrightarrow \boldsymbol{\psi_i}^{(k+1)} = ((\sum_u (\ \boldsymbol{\theta_u}^{(k+1)} \boldsymbol{\theta_u}^{(k+1)^T}\ )) + \lambda_i \mathbf{I_D}\ )^{-1} (\sum_u (\ [\mathbf{S}]_{u,i} \boldsymbol{\theta_u}^{(k+1)^T})) \tag{4.25}$$

Thus BCD updates to solve MF are given:

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

$$\forall (u,i) \in \mathcal{K} \ , \ k = 0, 1, ..., I_M \tag{4.26}$$

where $^{(k)}$ represents the index of the BCD iterations, (u,i) are the codebooks indexes at UE and BS, $[\mathbf{S}]_{u,i}$ 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 iteration reaches a predefined ϵ or a max number of iterations, I_M.

We have the following result: the sequence of updates $\{\boldsymbol{\theta_u}^{(k)}, \boldsymbol{\psi_i}^{(k)} \mid \forall (u,i) \in \mathcal{K}\}_k$ generated by BCD, in (4.26), is non-increasing (in k), and converges to a stationary point, as $k \to \infty$. Proof: see appendix A.

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 sub-problem (S1) is done by performing SGD for $f_1(\boldsymbol{\theta}_u) = \sum_{u\in\mathcal{U}_i} h_u(\boldsymbol{\theta}_u)$ in $(d.1)$, i.e., choosing at random a single index $u \in \mathcal{U}_i$ and computing the plain SGD $\widehat{\nabla f_1(\boldsymbol{\theta}_u)} = \widehat{\nabla} \left(\sum_{u\in\mathcal{U}_i} h_u(\boldsymbol{\theta}_u)\right) = h_u(\boldsymbol{\theta}_u)$, where u is a random index from $\mathcal{U}_i$, and $\widehat{\nabla f_1(\boldsymbol{\theta}_u)}$ is the plain SGD on $f_1()$ (in $e.1$). The update is then expressed as:

$$\boldsymbol{\theta_u}^{(k+1)} = \boldsymbol{\theta_u}^{(k)} - \alpha_k \widehat{\nabla f_1(\boldsymbol{\theta_u}^{(k)})} \ , = \boldsymbol{\theta_u}^{(k)} - \alpha_k \nabla h_u(\boldsymbol{\theta_u}^{(k)}) \ u \ \in \mathcal{U}_i \tag{4.27}$$

$$= \boldsymbol{\theta_u}^{(k)} + 2\alpha_k (\ (\sum_i ([\mathbf{S}]_{u,i} \boldsymbol{\psi_i}^{(k)})) - (\ (\sum_i \boldsymbol{\psi_i}^{(k)} \boldsymbol{\psi_i}^{(k)^T}) + \mu_u \mathbf{I_D} \ ) \boldsymbol{\theta_u}^{(k)}), \tag{4.28}$$

$u \ \in \mathcal{U}_i$, $k = 1..T$ 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 sub-problem (S2) is done by taking T plain SGD steps of $f_2(\boldsymbol{\psi}) = \sum_{i\in\mathcal{I}_u} h_i(\boldsymbol{\psi_i})$ in $d.2$), 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 sub-problem (S2) is expressed as:

$$\boldsymbol{\psi_i}^{(k+1)} = \boldsymbol{\psi_i}^{(k)} - \alpha_k \widehat{\nabla f_2(\boldsymbol{\psi_i}^{(k)})} = \boldsymbol{\psi_i}^{(k)} - \alpha_k \nabla h_2(\boldsymbol{\psi_i}^{(k)}) \ , \ i \in \mathcal{I}_\sqcap \tag{4.29}$$

$$= \boldsymbol{\psi_i}^{(k)} + 2\alpha_k (\ (\sum_u (\ [\mathbf{S}]_{u,i} \boldsymbol{\theta_u}^{(k)^T})) - (\ \sum_u (\ \boldsymbol{\theta_u}^{(k)} \boldsymbol{\theta_u}^{(k)^T} \ ) \ ) + \lambda_i \mathbf{I_D} \ ) \boldsymbol{\theta_u}^{(k)}), \tag{4.30}$$

$\forall i \in \mathcal{I}_u,\ \forall k = 1..T$ where i is a single index chosen randomly from $\mathcal{I}_u$,

$$\mathbf{t_i}^{(k)} = \sum_u (\ [\mathbf{S}]_{u,i} \boldsymbol{\theta_u}^{(k)^T}), \mathbf{P_i}^{(k)} = \sum_u (\ \boldsymbol{\theta_u}^{(k)} \boldsymbol{\theta_u}^{(k)^T}\ ) \tag{4.31}$$

where $\widehat{\nabla f_2(\boldsymbol{\psi_i})}$ is the plain SGD gradient calculated with respect to one sample $i \in \mathcal{I}_u$, chosen at random. We express the SGD MF updates as,:

$$\begin{cases} \boldsymbol{\theta_u}^{(k+1)} & = \boldsymbol{\theta_u}^{(k)} + 2\alpha_k (\ (\sum_i ([\mathbf{S}]_{u,i} \boldsymbol{\psi_i}^{(k)})) - (\ (\sum_i \boldsymbol{\psi_i}^{(k)} \boldsymbol{\psi_i}^{(k)^T}) + \mu_u \mathbf{I_D}\ ) \boldsymbol{\theta_u}^{(k)}) \\ \boldsymbol{\psi_i}^{(k+1)} & = \boldsymbol{\psi_i}^{(k)} + 2\alpha_k (\ (\sum_u (\ [\mathbf{S}]_{u,i} \boldsymbol{\theta_u}^{(k)^T})) - (\ (\ \sum_u (\ \boldsymbol{\theta_u}^{(k)} \boldsymbol{\theta_u}^{(k)^T}\ ) + \lambda_i \mathbf{I_D}\ ) \boldsymbol{\theta_u}^{(k)}) \end{cases} \tag{4.32}$$

$$\forall\ k = 0, 1, ..., T,\ \ i \in \mathcal{I}_u,\ \ u \in \mathcal{U}_i \tag{4.33}$$

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

BGD for MF (BGD MF):
Instead of having a closed form solution for each optimization block, BGD proceeds by taking T gradient steps, for each block T gradients steps. Therefore, the BGD updates for the MF problem are expressed as,

$$\begin{cases} \boldsymbol{\theta_u}^{(k+1)} & = \boldsymbol{\theta}_u^{(k)} + 2\alpha_k (\ (\sum_i ([\mathbf{S}]_{u,i} \boldsymbol{\psi_i}^{(k)})) - (\ (\sum_i \boldsymbol{\psi_i}_i^{(k)} \boldsymbol{\psi}_i^{(k)^T}) + \mu_u \mathbf{I_D}\ ) \boldsymbol{\theta_u}^{(k)}) \\ \boldsymbol{\psi_i}^{(k+1)} & = \boldsymbol{\psi_i}^{(k)} + 2\alpha_k (\ (\sum_u (\ [\mathbf{S}]_{u,i} \boldsymbol{\theta_u}^{(k)^T})) - (\ (\ \sum_u\ \boldsymbol{\theta_u}^{(k)} \boldsymbol{\theta_u}^{(k)^T}\ ) + \lambda_i \mathbf{I_D}\ ) \boldsymbol{\theta_u}^{(k)}) \end{cases} \tag{4.34}$$

$$\forall (u, i) \in \mathcal{K}\ ,\ \ k = 0, 1, ..., T, \tag{4.35}$$

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

4.4.2 BCD, BGD and BSGD solutions using Non-negative Matrix Factorization

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

BCD for NMF (BCD NMF):
The derivations of BCD for NMF (4.36), are identical to those of BCD for MF (4.26), followed by the corresponding projection operation $[]_+$. The updates of BCD for NMF derivations are given by:

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

$$\forall (u,i) \in \mathcal{K}\ ,\ k = 0, 1, ..., I_M \tag{4.36}$$

where $^{(k)}$ is the BCD iteration index, $[\mathbf{a}]_+ := max(\mathbf{a}, \mathbf{0})$ is applied element-by-element on $\mathbf{a}$, i.e., a Euclidean projection of $\mathbf{a}$ on $\mathbb{R}_+^D$. Since the projection is Euclidian (non-expansive operator), the Corollary stated in the previous subsection, proved in Annex A, 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,

$$\begin{cases} \boldsymbol{\theta_u}^{(k+1)} &= \Big[\boldsymbol{\theta_u}^{(k)} + 2\alpha_k(\ (\sum_i([\mathbf{S}]_{u,i}\boldsymbol{\psi_i}^{(k)})) - (\ (\sum_i \boldsymbol{\psi_i}^{(k)}\boldsymbol{\psi_i}^{(k)^T}) + \mu_u\mathbf{I_D}\ )\boldsymbol{\theta_u}^{(k)})\Big]_+ \\ \boldsymbol{\psi_i}^{(k+1)} &= \Big[\boldsymbol{\psi_i}^{(k)} + 2\alpha_k(\ (\sum_u(\ [\mathbf{S}]_{u,i}\boldsymbol{\theta_u}^{(k)^T})) - (\ \sum_u(\ \boldsymbol{\theta_u}^{(k)}\boldsymbol{\theta_u}^{(k)^T}\ )\ ) + \lambda_i\mathbf{I_D}\ )\boldsymbol{\theta_u}^{(k)})\Big]_+ \end{cases}$$

$$\forall\ k = 0, 1, ..., T,\ u \in \mathcal{U}_i,\ i \in \mathcal{I}_u \tag{4.37}$$

where u is a random index chosen from $\mathcal{U}_i$, i 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 derivations and solutions for BGD NMF are the same those of BGD MF, followed by a projection $[]_+$ i.e,

$$\begin{cases} \boldsymbol{\theta_u}^{(k+1)} &= \Big[\boldsymbol{\theta_u}^{(k)} + 2\alpha_k(\ (\sum_i([\mathbf{S}]_{u,i}\boldsymbol{\psi_i}^{(k)})) - (\ (\sum_i \boldsymbol{\psi_i}^{(k)}\boldsymbol{\psi_i}^{(k)^T}) + \mu_u\mathbf{I_D}\ )\boldsymbol{\theta_u}^{(k)})\Big]_+ \\ \boldsymbol{\psi_i}^{(k+1)} &= \Big[\boldsymbol{\psi_i}^{(k)} + 2\alpha_k(\ (\sum_u(\ [\mathbf{S}]_{u,i}\boldsymbol{\theta_u}^{(k)^T})) - (\ \sum_u(\ \boldsymbol{\theta_u}^{(k)}\boldsymbol{\theta_u}^{(k)^T}\ )\ ) + \lambda_i\mathbf{I_D}\ )\boldsymbol{\theta_u}^{(k)})\Big]_+ \end{cases}$$

$$\forall (u,i) \in \mathcal{K}\ ,\ k = 0, 1, ..., T, \tag{4.38}$$

where $[\mathbf{a}]_+ := \max(\mathbf{a}, \mathbf{0})$, $^{(k)}$ is the GD iterations 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.

4.5 Predictions for MF and NMF

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

$$\{\widehat{RSE}_{u,i} := (\widehat{\boldsymbol{\theta}}_u)^T\widehat{\boldsymbol{\psi}}_i \mid \forall (u,i) \in \mathcal{L}\} \tag{4.39}$$

where $\mathcal{L}$ is the test set and $\{\widehat{\boldsymbol{\theta}}_u)^T, \widehat{\boldsymbol{\psi}}_i\}$ 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:

$$(u^\star, i^\star) = \underset{(u,i)\in\mathcal{L}\cup\mathcal{K}}{argmax}\ \ (\widehat{\boldsymbol{\theta}}_u)^T\widehat{\boldsymbol{\psi}}_i. \tag{4.40}$$

4.6 Algorithm for the proposed Beam Alignment using MF/NMF

All the different methods proposed in this chapter, are neatly tied together and presented in Algorithm (4).

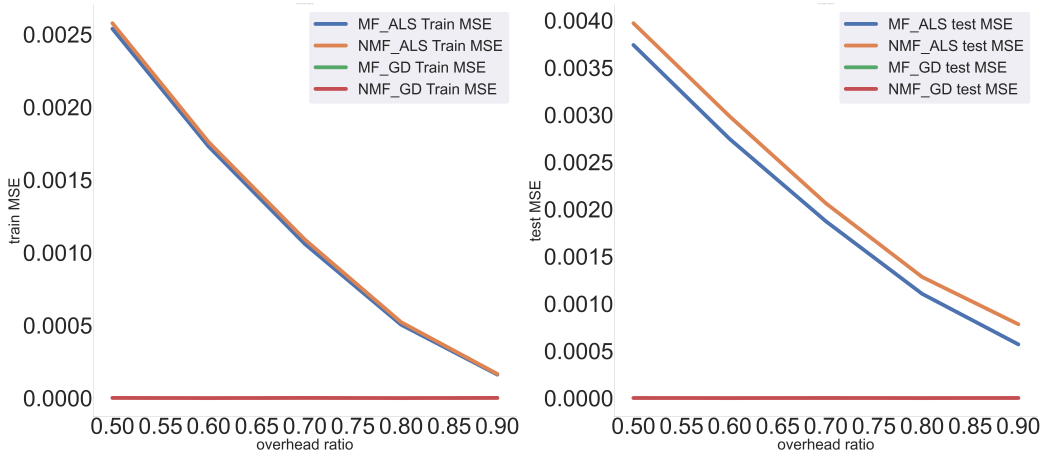
Algorithm 4 Proposed MF/NMF-based BA Method

Input: $\{\mathbf{f}_u\}_{\forall u \in \mathcal{T}}$, $\{\mathbf{w}_i\}_{\forall i \in \mathcal{R}}$, η
Generate randomly sub-sampled codebooks, $\mathcal{T}_{\mathcal{S}}, \mathcal{R}_{\mathcal{S}}$, satisfying $(|\mathcal{T}_{\mathcal{S}}| \times |\mathcal{R}_{\mathcal{S}}|)/(|\mathcal{T}| \times |\mathcal{R}|) = \eta$
Sound beam-pairs from the training set, $\mathcal{K} := \mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}$.
Record corresponding RSE and generate matrix $\mathbf{S}$ in (4.7).
Select the model: MF or NMF.
if MF model selected **then**
    Solve $(P1)$ with BCD for MF, in (4.26)
    or solve $(P1)$ with BGD for MF, in (4.35)
    or solve $(P1)$ with SGD for MF, in (4.33). At the end of training, return optimal latent vectors, $\{\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\psi}}_i\}_{(u,i)\in\mathcal{K}}$
end if
if NMF model selected **then**
    Solve $(P2)$ with BCD for NMF, in (4.36)
    or solve $(P2)$ with BGD for NMF, in (4.38)
    or solve $(P2)$ with SGD for NMF, in (4.37). At the end of training, return ideal latent vectors, $\{\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\psi}}_i\}_{(u,i)\in\mathcal{K}}$
end if
Use ideal latent vectors $\{\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\psi}}_i\}_{(u,i)\in\mathcal{K}}$ to predict unknown RSE of the test set, $\mathcal{L}$, in (4.39)
Search training and test sets for the beam-pair with the largest RSE, (4.40)
Output: $\mathbf{f}_{u^\star}$, $\mathbf{w}_{i^\star}$ =0

4.7 Numerical Simulations

This section outlines the experimental procedure, focusing on the continuous evaluation of Machine Learning model predictions in comparison to Exhaustive Search. The simulations encompass various setups, ranging from 16 to 1024 antennas at UE and BS, with specific emphasis on Massive MIMO configurations (256, 512, 1024). The overhead ratio spans from 0.1 to 0.9. The central challenge revolves around identifying the minimum overhead value that yields accurate predictions for the optimal Tx-Rx beam pair associated with the highest RSE.

Cross validation is a grid search process that aims to find the optimal hyperparameters of the proposed models (chosen regarding the lowest achieved test error); Tx-Rx Regularizers are 20 linearly-equispaced scalars in from 0.001 to 100, the number of used latent factors (MF model complexity) is 2,3,4 depending on the size of RSE Matrix. The learning rate (Gradient Descent step size) is fixed and is equal to



(a) 512×512 Train MSE in function of the overhead ratio

(b) 512×512 Test MSE in function of the overhead ratio

Figure 4.4: Train and Test MF/NMF Performance in function of the overhead ratio

0.01 for all the cross validation procedures. The rank of the MIMO channel which is the number of mmWave channel paths is fixed and is equal to 1 (Narrowband Line-of-Sight model).

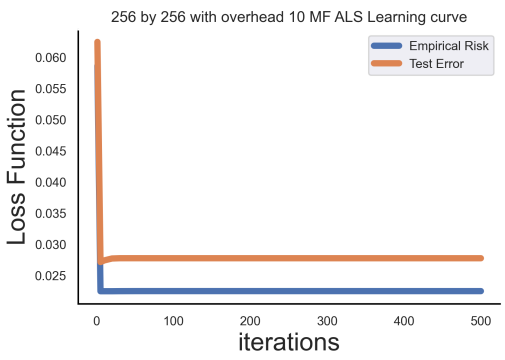
In our calculations for both MF and NMF, we employed two distinct algorithms:

- Alternating Least Squares (ALS): This method relies on equations derived from (4.26) and (4.36).
- Gradient Descent (GD): We utilized equations from (4.35) and (4.38) for this approach.

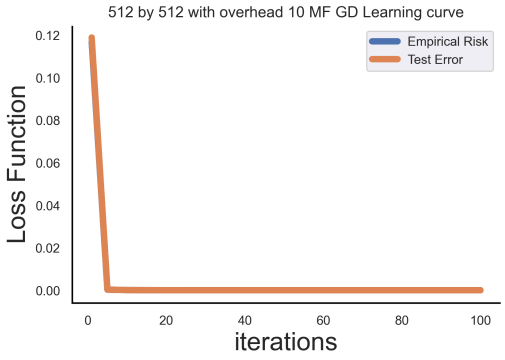
To comprehensively evaluate model performance, we considered four distinct models in total: MF ALS, MF GD, NMF ALS, and NMF GD (the totality of proposed models and variants including SGD for MF and NMF are simulated in the next chapter). For robustness and reliability, every component involved in the dataset RSE Matrix generation, such as the channel, beamformer, and Tx symbol, is generated randomly. As we work with increasingly large dataset matrix sizes, the process of fine-tuning hyperparameters becomes computationally expensive. This tuning phase is conducted offline. Once the optimal combination of hyperparameters is determined, we proceed with the prediction phase, which involves completing the matrix task.

4.7.1 Train Performance

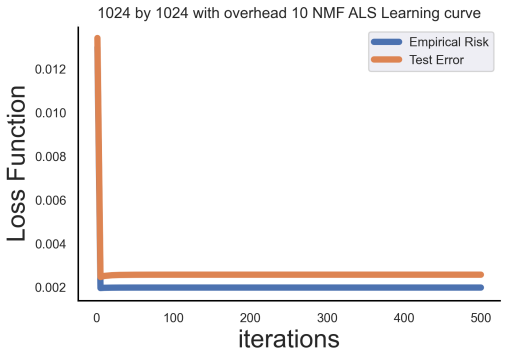
Matrix Factorization models require continuous fine-tuning to identify the optimal combinations of hyperparameters. This includes selecting the appropriate learning rate, determining the number of Latent Factors and tuning the regularization factors for both UE and BS components through multiple iterations. During training, we assess performance by monitoring the Mean Squared Error of the cost function applied to the training set samples in relation to the overhead ratio. In contrast,



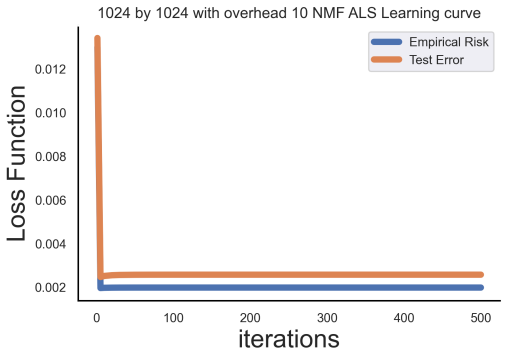
(a) 256 by 256 with overhead 20 MF ALS Learning curve



(b) 512 by 512 with overhead 20 MF GD Learning curve



(c) 1024 by 1024 with overhead 10 NMF ALS Learning curve



(d) 1024×1024 with overhead 0.1 NMF ALS Learning curve

Figure 4.5: MF/NMF Learning curves: Train/Test MSE in function of the learning iterations

prediction error measures the difference between the predicted RSE value (from MF) and the true RSE value obtained through Exhaustive BA. This provides a localized evaluation for a specific experimental instance (in all results tables, it indicates the prediction error of the best beam-pair instance.) Therefore, MSE serves as a more comprehensive, global and informative evaluation metric.

In Tables (4.3), (4.2), and (4.1), we monitor the training Mean Squared Error for the four proposed models across matrix sizes of 256×256, 512×512, and 1024×1024. This evaluation spans overhead ratios ranging from 0.9 to 0.5. Notably, there are distinct differences in error range and behavior between the Alternating Least Squares and Gradient Descent models in both Matrix Factorization and Non-Negative Matrix Factorization. This distinction becomes more evident when analyzing the curves that plot training MSE as a function of the overhead ratio.

For GD-based models, the error range remains relatively consistent at around $10e-08$, while ALS-based models exhibit a wider range of errors, typically around $10e-04$. In the case of both MF and NMF models, as depicted in Figure (4.4), Train MSE generally decreases with an increase in the overhead ratio, indicating that with fewer training samples available i.e., higher overhead ratios, the models tend to learn more effectively.

It's noteworthy that in all simulations, the training MSE remains extremely low, underscoring the fact that all models have effectively learned to make accurate predictions on the training samples across the proposed range of overhead values.

Model	overhead = 0.5	overhead = 0.6	overhead = 0.7	overhead = 0.8	overhead = 0.9
MF ALS	0.013973	0.005810	0.003596	0.001714	0.000516
NMF ALS	0.00857	0.00588	0.003652	0.001730	0.000534
MF GD	1.921554e-08	1.000214e-08	1.001488e-08	1.157753e-08	1.026492e-08
NMF GD	1.001505e-08	1.000156e-08	1.000399e-08	9.997341e-09	9.988570e-09

Table 4.1: 256 by 256 Train MSE in function of the overhead ratio

Model	overhead = 0.5	overhead = 0.6	overhead = 0.7	overhead = 0.8	overhead = 0.9
MF ALS	0.002538	0.001729	0.001059	0.000504	0.000161
NMF ALS	0.002576	0.001759	0.001089	0.000521	0.000165
MF GD	1.000311e-06	1.000200e-08	9.997494e-07	9.995999e-09	9.999849e-07
NMF GD	1.00010e-06	1.000122e-08	1.000090e-06	1.0004463e-08	1.0001752e-06

Table 4.2: 512 by 512 Train MSE in function of the overhead ratio

Model	overhead = 0.5	overhead = 0.6	overhead = 0.7	overhead = 0.8	overhead = 0.9
MF ALS	0.000680	0.000457	0.000279	0.000132	4.187750e-05
NMF ALS	0.000695	0.000467	0.000285	0.000137	4.365256e-05
MF GD	9.998334e-07	9.996394e-09	9.998379e-09	1.023600e-08	9.999563e-09
NMF GD	1.000130e-06	9.998424e-09	1.000207e-08	1.007870e-08	1.000090e-08

Table 4.3: 1024 by 1024 Train MSE in function of the overhead ratio

4.7.2 Test Performance

The analysis of Test Performance primarily relies on tracking the Mean Squared Error computed on the Test set samples, which represent the unknown entries involved

in the Matrix Completion task. In Tables (4.6), (4.5), and (4.4), we observe the Test MSE for the four proposed models across matrix sizes of 256×256, 512×512, and 1024×1024, while considering overhead ratios ranging from 0.9 to 0.5.

Once again, we notice that there are clear distinctions in error range and behavior between the Alternating Least Squares and Gradient Descent models, both in Matrix Factorization and Non-Negative Matrix Factorization. These differences are reflected not only in the absolute error values but also in the trends observed when plotting the Test MSE as a function of the overhead ratio, as shown in Figure (4.5). These results parallel the patterns observed in the Train Performance analysis for each proposed setup.

Furthermore, it is worth highlighting that, for both MF and NMF, the performance of ALS calculations stands apart from that of GD, and the ranges of error values exhibit notable distinctions.

Model	overhead = 0.5	overhead = 0.6	overhead = 0.7	overhead = 0.8	overhead = 0.9
MF ALS	0.014447	0.008402	0.005680	0.003304	0.001416
NMF ALS	0.012085	0.008819	0.005966	0.003542	0.001937
MF GD	2.310921e-08	1.020069e-08	1.020804e-08	1.226321e-08	1.069049e-08
NMF GD	1.022159e-08	1.019264e-08	1.019379e-08	1.018302e-08	1.014422e-08

Table 4.4: 256 by 256 Test MSE in function of the overhead ratio

Model	overhead = 0.5	overhead = 0.6	overhead = 0.7	overhead = 0.8	overhead = 0.9
MF ALS	0.003739	0.002744	0.001872	0.001104	0.000568
NMF ALS	0.003970	0.002983	0.002062	0.001282	0.000782
MF GD	1.000053e-06	1.012991e-08	1.000072e-06	1.011945e-08	1.000348e-06
NMF GD	9.998704e-07	1.011856e-08	1.000473e-06	1.012624e-08	1.000475e-06

Table 4.5: 512 by 512 Test MSE in function of the overhead ratio

Model	overhead = 0.5	overhead = 0.6	overhead = 0.7	overhead = 0.8	overhead = 0.9
MF ALS	0.000994	0.000717	0.000480	0.000295	0.000159
NMF ALS	0.001050	0.000768	0.000525	0.000328	0.000243
MF GD	9.999064e-07	1.004559e-08	1.005048e-08	1.031886e-08	1.005227e-08
NMF GD	1.000179e-06	1.004392e-08	1.005100e-08	1.015194e-08	1.004839e-08

Table 4.6: 1024 by 1024 Test MSE in function of the overhead ratio

4.7.3 Train/Test Performance

The Train/Test performance analysis involves monitoring both Train and Test Mean Squared Error over a series of 100 iterations, which is depicted in the learning curve shown in Figure (4.5). Notably, ALS calculations, whether for Matrix Factorization or Non-Negative Matrix Factorization, demonstrate rapid convergence. Right from the initial iterations, the ALS-based models quickly adjust their parameters, leading to a substantial reduction in both Test and training MSE, nearly approaching zero. On the other hand, the Gradient Descent method necessitates more iterations to achieve a similar drop in MSE. In terms of Quality of Service, Table (4.7) provides an overview of the minimum overhead necessary for the proposed setups, the optimal model, i.e. the one with the lowest Test error, and the associated error values

(Training MSE, Test MSE, and prediction distance for the best beam-pair instance). Remarkably, for a Line-of-Sight narrowband scenario, massive MIMO configurations demonstrate that only 10% of beam pairs at UE and BS are needed to conduct complete Beam Alignment while maintaining highly accurate predictions.

Matrix Size	Optimal Model	Min Overhead	Train MSE	Test MSE	Prediction Error
16 by 16	NMF GD	0.7	1.663786e-06	0.158286	0.048528
32 by 32	NMF GD	0.6	0.000525	0.000328	0.000243
64 by 64	MF GD	0.3	1.005048e-08	1.031886e-08	1.005227e-08
128 by 128	NMF GD	0.1	8.042019e-06	1.540287e-05	9.044893e-05
256 by 256	MF GD	0.1	1.722457e-06	2.00182e-06	0.001485
512 by 512	NMF GD	0.1	3.184227e-07	3.534617e-07	0.000191
1024 by 1024	NMF GD	0.1	1.002663e-08	1.0099426e-08	0.000118

Table 4.7: The minimum overhead required for the proposed configurations

Table (4.7) clearly demonstrates that larger matrices provide more information for the models to learn from, ensuring higher prediction quality. In summary, Matrix Factorization tools proves its ability to train, cross-validate hyperparameters, and make accurate predictions across a fully-analog low-complexity system architecture.

4.8 Conclusion

In this chapter, our first contribution was presented. It begins with the theoretical foundation, offering a comprehensive insight into the proposed system model, which is followed by the formulation of the problem for each variant of Matrix Factorization. Subsequently, the chapter delves into the solutions, providing detailed derivations and theoretical proofs of the loss function's convergence. The final part of the chapter encompasses the algorithm's description and the presentation of numerical results. In addition to the theoretical convergence guarantees of the proposed algorithm, these results serve as concrete evidence of the success of our approach, which combines model-based and data-driven methods to address the challenge of large signaling overhead. Remarkably, our method achieves its objectives using only a mere 10% of the beams, making it a fully blind solution.

Now, having established the efficacy of our proposed solution for the Beam Alignment problems discussed in this chapter, we are poised to subject our Matrix Factorization techniques and their variants to more intricate experimental setups and an expanded system architecture. Furthermore, we aim to evaluate their performance in comparison to a shallow feed-forward neural architecture, with a specific focus on Quality of Service considerations. This endeavor represents a pivotal step in our research journey, as it pushes the boundaries of our methods and seeks to uncover their potential in increasingly complex scenarios before adding the practical quantization constraints later.

Chapter 5

Multi Layer Perceptron for blind and partial Beam Alignment in massive mmWave MIMO

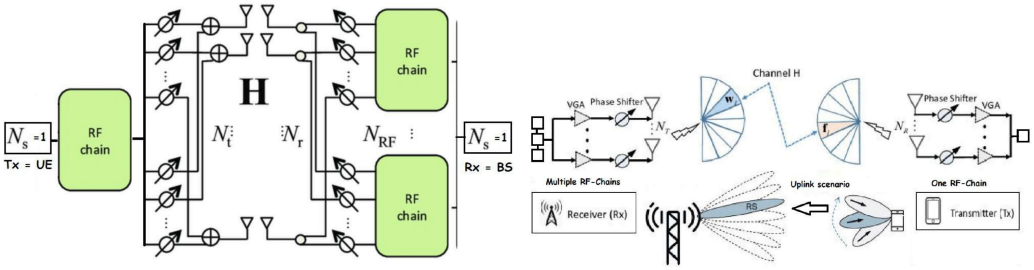
"Always remember that when it comes to markets, past performance is not a good predictor of future returns - looking in the rear- view mirror is a bad way to drive. Machine learning, on the other hand, is applicable to datasets where the past is a good predictor of the future."

Francois Chollet

5.1 Introduction

This chapter represents the second contribution of the thesis, employing both Matrix Factorization and Multi-Layer Perceptron to address a partial and blind Beam Alignment challenge within an extended system model and a more intricate experimental setup. While the previous chapter served to validate our approach within a basic system configuration, focusing primarily on assessing its feasibility and numerical success, the aim here is to expand the Uplink architecture. This expansion involves the addition of more RF chains and increased complexity at the base station, providing the necessary computational resources and meeting the associated requirements. We delve into the complexities of a wideband Non-Line-of-Sight channel, which presents a more demanding experimental environment. We employ Discrete Fourier Transform codebooks for beam patterns at both UE and BS instead of the uniform codebooks, used in the previous chapter. Additionally, as we explore new models, we investigate a wider range of Machine Learning tools that naturally align with the task at hand.

The formulation of the partial and blind BA problem, as well as our proposed solution, remains largely consistent with the previous chapter. The fundamental challenge remains completing a sparse and low-rank Received Signal Energy dataset



(a) Fully-analog MIMO architecture using a single RF Chain at UE and multiple RF Chains at BS

(b) Uplink Beam Alignment problem simplified illustration

Figure 5.1: Proposed BA diagram representation

matrix by solving a non-linear regression problem. However, this chapter introduces some key extensions and the new ML tool: the Multi-Layer Perceptron. The problem formulation and solution approach remain the same for both MLP and MF. To avoid redundancy, we do not reiterate the MF equations in this chapter (all MF and NMF derivations in the previous chapter hold here and all BCD, BGD, $BGSD$ related methods are used in this chapter). Finally, we present a comparative study between the two proposed learning models, exploring the trade-off between complexity and accuracy across various transmitted power regimes.

5.2 Point-to-point system architecture with one-RF chain at UE and multiple RF-chains at BS:

In this section, we illustrate the mmWave MIMO point-to-point wideband system model. We consider an Uplink transmission from a multiple-antenna user equipment UE using a single radio-frequency chain and a multiple-antenna base station BS using multiple radio-frequency chains, both wishing to align the most optimal fully-analog beamformer and combiner pair to establish a reliable initial link. Note that, the proposed ML methods are performed at the BS which has higher computational resources than the UE. Recall the signaling overhead Ω, defined as the total number of pilots needed for BA, and T denotes the total number of time slots in the frame. Thus, the effective rate r is expressed as,

$$r = (1 - \frac{\Omega}{T}) \log_2(1 + \text{SNR}) \tag{5.1}$$

Figure (5.1a) and (5.1b) provide the diagram representation of the proposed architecture. It consists on a simplified illustration of the proposed beam-sounding procedure for an Uplink scenario. The UE and BS are respectively equipped with Uniform Linear Arrays of N_T and N_R antenna.

5.2.1 Beam former and combiner

We propose a low-complexity fully-analog architecture where the UE has one radio-frequency chain and BS have N_{rf} radio-frequency chains. Recall that the number of RF chains at both UE and BS is theoretically assumed to be much smaller that the corresponding number of antennas. The UE selects its analog beamformer $\boldsymbol{f_u} \in \mathbf{C}^{N_T}$ from a DFT codebook of feasible beams choices, $u \in \mathcal{T}$, where $\mathcal{T}$ is the corresponding index set. Moreover, the BS selects its analog combiner $\mathbf{W_i} \in \mathbf{C}^{N_R \times N_{rf}}$ from a DFT codebook $i \in \mathcal{R}$ with $\mathcal{R}$ the index set of the codebook. The resulted beams from using DFT codebooks are equispaced. We denote by C_T the number of possible beamforming vectors at the UE, i.e. the size/cardinality of the UE codebook, $|\mathcal{T}| = C_T$. Similarly, we consider C_R as the number of possible combining vectors at the BS, i.e. the size/cardinality of the BS codebook, $|\mathcal{R}| = C_R$. Both beamforming and combining are fully done in the analog domain using phase-shifters at UE and BS, thus they satisfy the following constant modulus constraints:

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

$$\boldsymbol{W}_i \in \mathbf{C}^{N_R \times N_{rf}} \ , \ | \ [\ \boldsymbol{W}_i \ ]_{r,t} \ | = \frac{1}{N_{rf} N_R} \ , \ \forall \ r \in \{1, ..., N_R\}, \ \forall \ t \in \{1, ..., N_{rf}\}$$

Besides, the processing among an Uplink transmission is all done at the BS. For our proposed approach, the BS is responsible of receiving signal energies, 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 so that they establish a reliable transmission link.

5.2.2 Wideband Saleh-Valenzuela mmWave Channel model

We adopt the wideband Saleh-Valenzuela [22] channel model $\mathbf{G} \in \mathbf{C}^{N_R \times N_T}$ given by

$$\mathbf{G}(k) = \sqrt{\frac{1}{N_c}} \sum_{l=1}^{N_c} \mathbf{H}_l e^{-j2\pi lk/N_c}, \ \forall \ k \in \{1, ..., N_C\} \tag{5.2}$$

where Nc represents the number of sub-carriers over the whole bandwidth through an $OFDM$ scenario, k is the index of sub-carrier k, and $\mathbf{H}_l \in \mathbf{C}^{N_R \times N_T}$ is the narrow band channel model representing the time domain channel impulse response with L-tapped delays given by,

$$\mathbf{H}_l = \sqrt{\frac{N_T N_R}{L}} \sum_{i=1}^{L} \rho_i \ \mathbf{a}_R(\theta_i^{(R)}) \mathbf{a}_T^H(\theta_i^{(T)}) \tag{5.3}$$

where L is number of paths (rank) of the channel, $\theta_i^{(R)}$ and $\theta_i^{(T)}$ are angles of arrival at the BS and angles of departure from the UE, noted (AoA/AoD) corresponding 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 $\rho_i \sim \mathcal{CN}(0, 1), \ \forall i$. Last but not least,

$\mathbf{a}_R(\theta_i^{(R)}) \in \mathbf{C}^{N_R}$ and $\mathbf{a}_T(\theta_i^{(T)}) \in \mathbf{C}^{N_T}$ are the array response vectors at both the UE and BS, respectively. We consistently assume that the channel is static and completely unknown to both UE and BS.

5.2.3 Received Signal Energies

We denote beam pair (u, i) as the combination of UE beamformer indexed u from the UE codebook $\mathcal{T}$, and combiner indexed i in the BS codebook $\mathcal{R}$. Again, the signal at the BS resulting from applying beam pair (u, i), $\mathbf{y}_{u,i} \in \mathbf{C}^{N_{rf}}$ is expressed as,

$$\mathbf{y}_{u,i} = \mathbf{W}_i{}^H \mathbf{G}(k)\, \boldsymbol{f}_u s_u + \boldsymbol{n}_i,\ \forall\ (u,i) \in \mathcal{T} \times \mathcal{R}, \tag{5.4}$$

where $s_u = \sqrt{P_u}$ is the transmitted pilot symbol associated with $\boldsymbol{f}_u$ (having power P_u) over $\boldsymbol{n}_i = \mathbf{W}_i^H \mathbf{n}$, the effective additive white Gaussian noise $AWGN$ with unit variance ($\sigma^2 = 1$). We propose to send one symbol in order to maximize SNR at BS via array gain. To that end, we define the received Signal to Noise Ratio for the beam-pair (u, i) as

$$\mathrm{SNR}_{u,i} = P_u ||\mathbf{W}_i^H \mathbf{G}(k) \boldsymbol{f}_u||_2^2,\ \forall\ (u,i) \in \mathcal{T} \times \mathcal{R} \tag{5.5}$$

We assume a fully-blind approach, i.e., neither the BS nor the UE have any knowledge of $\mathbf{G}$. Thus, computing the above SNR expression is not feasible. Similarly to the previous chapter, we will approximate the SNR of beam-pair (u, i) in (6.2) using the corresponding instantaneous Received Signal Energies, expressed as,

$$\mathrm{RSE}_{u,i} = ||\mathbf{y}_{u,i}||_2^2,\ \forall (u,i) \in \mathcal{T} \times \mathcal{R}. \tag{5.6}$$

In other words, we will assume that $RSE_{u,i} \approx \mathrm{SNR}_{u,i}$, for each beam-pair $(u, i) \in \mathcal{T} \times \mathcal{R}$.

5.3 Problem formulation

5.3.1 Problem statement for MLP

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. We denote by $\{\mathbf{z_2}, ..., \mathbf{z_{J-1}}\}$ all the hidden layer. We assume for simplicity that width of all the layer is the same, denoted as D, i.e., $\{\mathbf{z_2} \in \mathbb{R}^D, ..., \mathbf{z_{J-1}} \in \mathbb{R}^D\}$; see Fig 5.2. The eqt describing layer 1 is given by:

$$\mathbf{z_1} = \sigma_1(\phi_\mathbf{1} z_0) = \sigma_1(\phi_\mathbf{1} 1) \tag{5.7}$$

where $\mathbf{z_1} \in \mathbb{R}^D$ is the output of layer 1, $\phi_\mathbf{1} \in \mathbb{R}^D$ the resulting weight vector, $\sigma_1() : \mathbb{R} \longrightarrow \mathbb{R}^D$ is the non-linear activation function for layer 1. We use a one hot encoding for the MLP input $z_0 \in \mathbb{R}$, i.e., $z_0 = 1$ for all training samples,

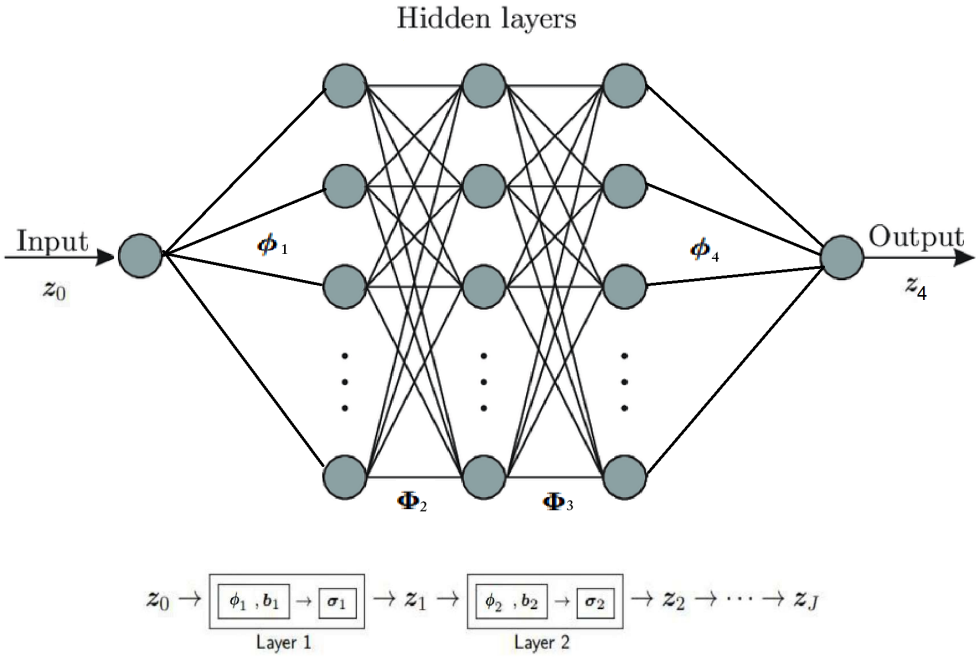


Figure 5.2: Multi Layer Perceptron Architecture (Toy example with $J = 4$)

$\forall (u,i) \in \mathcal{K}$. We express the output of the hidden layers, $\{\mathbf{z_j} \in \mathbb{R}^D\}_{j=2}^{J-1}$ as, $\mathbf{z_j} = \sigma_j(\mathbf{\Phi_j z_{j-1}})$, $\forall\ j \in \{2, ..., J-1\}$ where $\mathbf{z_{j-1}} \in \mathbb{R}^D$ is the input of layer j and $\mathbf{z_j} \in \mathbb{R}^D$ its output , $\forall\ j \in \{2, ..., J-1\}$, $\mathbf{\Phi_j} \in \mathbb{R}^{D \times D}$ the weight matrix for layer j , $\forall\ j \in \{2, ..., J-1\}$, and $\sigma_{j-1}() : \mathbb{R}^D \longrightarrow \mathbb{R}^D$ the element-by-element non-linear activation function for layer j, $\forall\ j \in \{2, ..., J-1\}$.

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

$$z_J := \sigma_J(\phi_{\mathbf{J}} ... \sigma_2(\mathbf{\Phi_2}(\sigma_1(\phi_{\mathbf{1}})))) \tag{5.8}$$

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 MSE loss $l_{u,i}$ for sample (u,i) in training set $\mathcal{K}$, as the distance between the MLP output z_J, and the known RSE label for beam-pair (u,i), $\text{RSE}_{u,i}$:

$$l_{u,i} := (z_J - RSE_{u,i})^2 = (\ \underbrace{\sigma_J(\phi_{\mathbf{J}} ... \sigma_2(\mathbf{\Phi_2}(\sigma_1(\phi_{\mathbf{1}}))))}_{\text{MLP output}} - \underbrace{RSE_{u,i}}_{\text{RSE value}}\ )^2\ ,\ \forall (u,i)\ \in \mathcal{K}$$

Consequently, the empirical risk is defined as the average of 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 $(P3)$.

$$(P3) := \{(\phi_{\mathbf{1}}{}^*, \mathbf{\Phi_2}^*, ..., \phi_{\mathbf{J}}{}^*) \begin{cases} \underset{\phi_{\mathbf{1}}, \mathbf{\Phi_2}, ..., \mathbf{\Phi_{J-1}}, \phi_{\mathbf{J}}}{argmin} \ \frac{1}{|\mathcal{K}|} \sum_{(u,i)\in\mathcal{K}} \ l_{u,i}(\phi_{\mathbf{1}}, \mathbf{\Phi_2}, ..., \mathbf{\Phi_{J-1}}, \phi_{\mathbf{J}}) \\ \phi_{\mathbf{1}} \ \in \ \mathbb{R}^D, \mathbf{\Phi_2} \ \in \ \mathbb{R}^{D\times D}, ..., \ \mathbf{\Phi_{J-1}} \ \in \ \mathbb{R}^{D\times D}, \ \phi_{\mathbf{J}} \ \in \ \mathbb{R}^{1\times D} \end{cases} \tag{5.9}$$

5.3.2 Proposed solution

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:

$$l_{\mathcal{B}} := \frac{1}{|\mathcal{B}|} \sum_{u,i\in\mathcal{B}} (\sigma_J(\phi_{\mathbf{J}}...\sigma_2(\mathbf{\Phi_2}(\sigma_1(\phi_{\mathbf{1}})))) - RSE_{u,i})^2 \ , \ \forall \ (u,i) \in \mathcal{B} \tag{5.10}$$

We express the partial derivative of the loss corresponding to the mini-batch $l_{\mathcal{B}}$, w.r.t. each layer $\mathbf{\Phi_j}, j\{1, ..., J\}$ as:

$$\frac{\partial l_{\mathcal{B}}}{\partial \mathbf{\Phi_j}} = \frac{1}{|\mathcal{B}|} \sum_{(u,i)\in\mathcal{B}} (\ \delta_j \mathbf{z}_{j-1}{}^T \ ), \ \forall j \in \{1, ..J\} \ , \tag{5.11}$$

where:

$$\delta_j \triangleq \begin{cases} (\mathbf{\Phi_{j+1}}{}^T \delta_{j+1}) \circ \sigma_j{}', j < J \\ 2(z_J - RSE_{u,i}) \circ \sigma_j{}', j = J \ , \ (u,i) \in \mathcal{B} \end{cases} , \sigma_j{}' \triangleq \frac{\partial \sigma(u)}{\partial u} = [\frac{\partial \sigma(u_1)}{\partial u_1}, ..., \frac{\partial \sigma(u_{d_j})}{\partial u_{d_j}}]^T , \tag{5.12}$$

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

$$\mathbf{\Phi_j^{(k+1)}} = \mathbf{\Phi_j^{(k)}} - \beta_j{}^{(k)} \frac{\partial l_{\mathcal{B}}}{\partial \mathbf{\Phi_j}}\Big|_{\mathbf{\Phi_j^{(k)}}}, \forall j \in \{1, ..., J\}, \ k = \{1, ..., T\} \tag{5.13}$$

where $^{(k)}$ in the BP iteration index, $\mathbf{\Phi_j^{(k)}}$ is value of $\mathbf{\Phi_j}$ at iteration k, $\beta_j{}^{(k)}$ BP step-size (learning rate) for layer j at iteration k, and $\frac{\partial l_{\mathcal{B}}}{\partial \mathbf{\Phi_j}}|_{\mathbf{\Phi_j^{(k)}}}$ the partial derivative given in (5.11) evaluated at $\mathbf{\Phi_j^{(k)}}$.

5.3.3 Back-propagation Algorithm with mini-batch:

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

1. Compute the loss function $l_{\mathcal{B}}$, for all samples in mini-batch $(u,i) \in \mathcal{B}$, in (5.10).
2. Compute partial derivative $\frac{\partial l_{\mathcal{B}}}{\partial \mathbf{\Phi_j}}$ of mini-batch loss l_B, w.r.t. $\mathbf{\Phi_j}$, in (5.11).
3. Update Weights of each layer, as in (5.13).

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

5.3.4 Prediction using MLP:

The MLP prediction for sample (u,i) in the test set $\mathcal{L}$, using optimal weights $\phi_{\mathbf{1}}{}^*, \mathbf{\Phi_2}{}^*, ..., \phi_{\mathbf{J}}{}^*$:

$$\hat{z_J} = \sigma_J(\phi_{\mathbf{J}}{}^* ... \sigma_2(\mathbf{\Phi_2}{}^*(\sigma_1(\phi_{\mathbf{1}}{}^*)))), \forall (u,i) \in \mathcal{L} \tag{5.14}$$

Therefore, the Test MSE is defined as

$$\frac{1}{|\mathcal{L}|} \sum_{(u,i)\in\mathcal{L}} \left(\widehat{RSE_{u,i}} - \sigma_J(\phi_{\mathbf{J}}{}^* ... \sigma_2(\mathbf{\Phi_2}{}^*(\sigma_1(\phi_{\mathbf{1}}{}^*)))) \right)^2 \tag{5.15}$$

. We then select the optimal indexes $u^\star$ and $i^\star$ related to the highest $RSE_{u,i}$ value:

$$(u^\star, i^\star) = argmax_{(u,i)\in\mathcal{L}\cup\mathcal{K}} \ \{RSE_{u,i} | \forall (u,i) \in \mathcal{K}\} \cup \{\hat{RSE}_{u,i} | \forall (u,i) \in \mathcal{L}\} \tag{5.16}$$

5.4 Algorithms of the proposed Beam Alignment using MLP and MF/NMF

The algorithm for MLP based Beam Alignment is given by (5):

Algorithm 5 Proposed MLP-based BA Method

Input: $\{\mathbf{f}_u\}_{\forall u\in\mathcal{T}}$, $\{\mathbf{W}_i\}_{\forall i\in\mathcal{R}}$, η, Pu
- Generate randomly sub-sampled codebooks, $\mathcal{T_S}, \mathcal{R_S}$, satisfying $(|\mathcal{T_S}|.|\mathcal{R_S}|)/(|\mathcal{T}|\times|\mathcal{R}|) = \eta$
- 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 (4.7)
- Train MLP weights (using Back-Propagation algorithm)
  return optimal weights, $\{\phi_{\mathbf{1}}{}^*, \mathbf{\Phi_2}{}^*, ..., \phi_{\mathbf{J}}{}^*\}$
- Use optimal parameters $\{\phi_{\mathbf{1}}{}^*, \mathbf{\Phi_2}{}^*, ..., \phi_{\mathbf{J}}{}^*\}$, to predict unknown RSE of test set, $\mathcal{L}$, in (5.15)
- Search training and test sets, for optimal beam-pair $(u^\star, i^\star)$, holding the largest RSE, (5.16)

Output: $\boldsymbol{f}_{u^\star}$, $\mathbf{W}_{i^\star}$

On the other hand, as we mentioned in the introduction of this chapter, we aim to apply MF/NMF on the same dataset (regarding the updated wideband system model) in order to compare its performance to MLP based approach. The MF/NMF formulated problems (P1) and (P2) from the previous chapter and the corresponding proposed solutions using MF and variants hold here. The dataset $\mathbf{S}$ coefficients are numerically different (due to a different setup). However, the input-output equations remain the same. Therefore, we skip re-writing them in this chapter. Due to the fact that the updates are given in close-form solution, we can quantify the computational complexity of the corresponding MF models. As seen from the updates for BCD MF and BCD NMF, we have to invert two $D \times D$ matrices (for sum-problems S1 and S2). Thus the computational complexity per-iteration 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 (for 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 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 (6).

Algorithm 6 Proposed MF/NMF-based BA Method

Input: $\{\mathbf{f}_u\}_{\forall u\in\mathcal{T}}$, $\{\mathbf{W}_i\}_{\forall i\in\mathcal{R}}$, η, Pu
- Generate randomly sub-sampled codebooks, $\mathcal{T}_{\mathcal{S}}, \mathcal{R}_{\mathcal{S}}$, satisfying $(|\mathcal{T}_{\mathcal{S}}|.|\mathcal{R}_{\mathcal{S}}|)/(|\mathcal{T}|\times|\mathcal{R}|) = \eta$
- Sound beam-pairs from training set, $\mathcal{K} := \mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}$.
- Record corresponding RSE in and generate mat. $\mathbf{S}$, in (4.7)
- Select model: MF or NMF
- **IF** MF model selected
  solve $(P1)$ with BCD for MF, in (4.26) or solve $(P1)$ with BGD for MF, in (4.35) or solve $(P1)$ with SGD for MF, in (4.33). At the end of training, return optimal latent vectors, $\{\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\psi}}_i\}_{(u,i)\in\mathcal{K}}$
- **IF** NMF model selected
  solve $(P2)$ with BCD for NMF, in (4.36) or solve $(P2)$ with BGD for NMF, in (4.38) or solve $(P2)$ with SGD for NMF, in (4.37). At the end of training, return ideal latent vectors, $\{\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\psi}}_i\}_{(u,i)\in\mathcal{K}}$
- Use ideal latent vectors $\{\widehat{\boldsymbol{\theta}}_u, \widehat{\boldsymbol{\psi}}_i\}_{(u,i)\in\mathcal{K}}$, to predict unknown RSE of test set, $\mathcal{L}$, in (4.39)
- Search training and test sets, for beam-pair w/ largest RSE, (4.40)

Output: $\mathbf{f}_{u^\star}$, $\mathbf{W}_{i^\star}$

While for MF BCD and NMF BCD the only hyper-parameter is the model size D, however MF BGD and NMF BGD require in addition to D, α^k the GD step-size as hyper-parameters.

5.5 Numerical simulations and comparison

This section illustrate the experimental protocol. 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 hyper-parameters 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.

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

Table 5.1: Point-to-point BA: proposed system parameters and hyperparameters

5.5.1 MF/NMF training and test QoS Performance

We propose to look into six models in total (BCD MF, BCD NMF, BGD MF, BGD NMF, SGD MF, SGD NMF) w.r.t three transmitted power regimes: high $Pu = 1W$, medium $Pu = 10^{-1}W$ and low $Pu = 10^{-2}W$ with fixed $\sigma^2 = 1$. In table (5.1), we give a complete summary for the proposed system-parameters. We use the training Normalized MSE to evaluate the training error, expressed as:

$$\text{Train NMSE} = \frac{1}{|\mathcal{K}|}\Big(\sum_{(u,i)\in\mathcal{K}}\Big(\frac{\widehat{\boldsymbol{\theta}_{\boldsymbol{u}}^T}\widehat{\boldsymbol{\psi}_i} - RSE_{u,i}}{RSE_{u,i}}\Big)^2\Big) \tag{5.17}$$

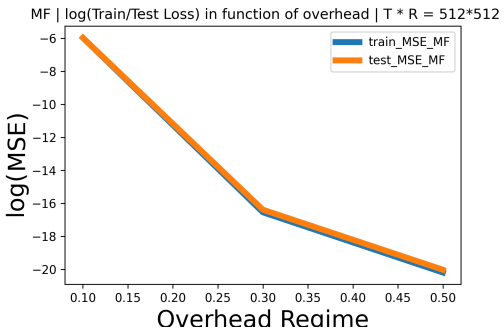
We similarly define:

$$\text{Test NMSE} = \frac{1}{|\mathcal{L}|}\Big(\sum_{(u,i)\in\mathcal{L}}\Big(\frac{\widehat{RSE_{u,i}} - \widehat{\boldsymbol{\theta}_{\boldsymbol{u}}^T}\widehat{\boldsymbol{\psi}_i}}{\widehat{RSE_{u,i}}}\Big)^2\Big) \tag{5.18}$$

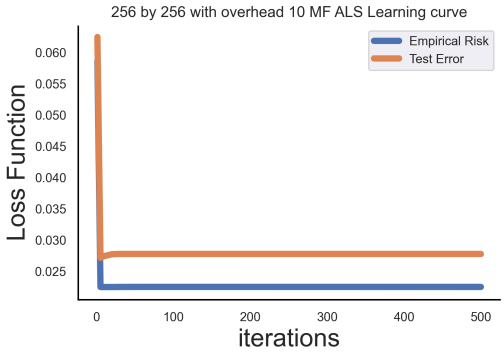
The behavior of BCD-based models differs significantly from GD-based models, presenting distinct characteristics in both MF and NMF:

- Error Range: BCD-based models exhibit a distinct range of training errors compared to GD models. Specifically, BCD models have an error range of approximately 10^{-4}, whereas GD models achieve a lower error range of about 10^{-7}. This suggests that GD methods are more accurate in minimizing error.

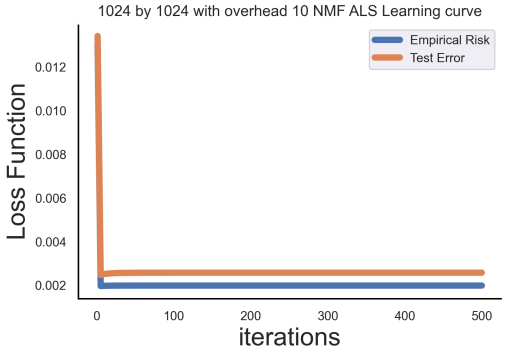
- Convergence Speed: BCD models, on the other hand, showcase a remarkable trait: they converge faster. These models quickly reduce the cost function to low values right from the initial iterations. In contrast, GD models may take longer to reach their optimal solutions but do so with a high level of precision.



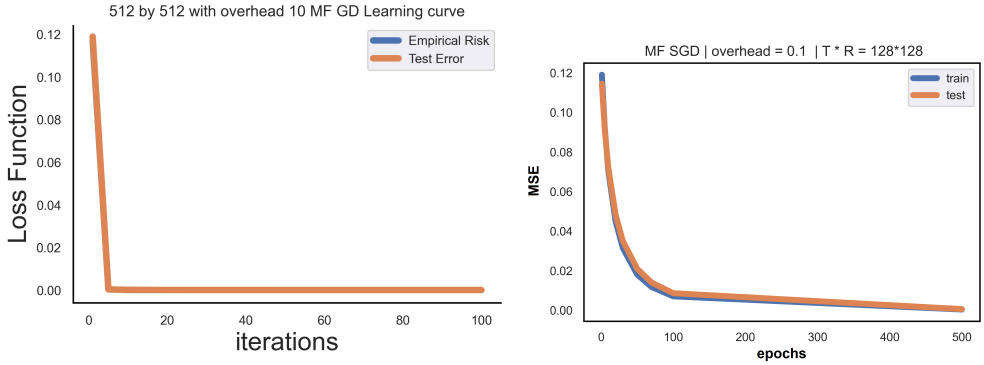
(a) 512×512 Train/Test loss in function of the overhead ratio



(b) Learning curve: 256 × 256 with overhead 0.1 $BCDMF$



(c) Learning curve: 1024×1024 with overhead 0.1 $BCDNMF$



(a) Learning curve: 512×512 with overhead 0.1 $BGDMF$

(b) Learning curve: 128×128 with overhead 0.1 $BCDSGD$

Figure 5.4: MF/NMF Train/Test performance and Learning curves

MIMO setup	Optimal hyperparameters	Min Overhead	Train NMSE	Test NMSE
128 by 128	BGD NMF{D=2, (λ,μ)=(0.0001,0.0001), α_k=0.001}	0.1	8.407746e-06	9.147875e-06
256 by 256	BGD MF{D=3, (λ,μ)=(0.0001,0.0001), α_k=0.001}	0.1	4.102708e-06	7.344720e-06
512 by 512	BGD MF{D=4, (λ,μ) =(0.0001,0.0001), α_k=0.001}	0.1	8.374633e-07	9.417057e-07
1024 by 1024	SGD NMF{D=4, (λ,μ)=(0.0001,0.0001), α_k=0.01}	0.1	1.219227e-07	1.616363e-07

Table 5.2: MF/NMF — QoS Minimum overhead required for $P_u = 1W$

Additionally, when examining the behavior of MF and NMF models, it becomes evident that the Train $NMSE$ decreases as the overhead ratio η increases, as depicted in Figure (5.3a). This suggests that having a higher number of training samples, represented by a higher overhead ratio, contributes to reducing prediction errors.

It's worth noting that low and medium P_u regimes, characterized by noisy links between the UE and BS, represent a more challenging experimental environment. In this context, BCD-based models tend to excel in terms of speed, quickly reaching low error values. On the other hand, GD-based models, including $BSGD$, are known for their accuracy, effectively improving prediction quality compared to standard BGD methods.

In summary, the choice between BCD and GD methods depends on the specific requirements of the problem, balancing the need for fast convergence with the pursuit of highly accurate predictions, particularly in challenging, noisy communication environments.

Concerning the simulation figures for MF/NMF, the following sub-figures offer additional insights into the behavior of these models:

- Sub-figure 5.3a demonstrates the relationship between the overhead ratio and the train/test Normalized Mean Squared Error. It highlights that increasing the number of training samples results in a reduction in prediction errors, emphasizing the importance of a larger dataset for improved model performance.

MIMO setup	Optimal hyperparameters	Min Overhead	Train NMSE	Test NMSE
128 by 128	SGD NMF {D=2, (λ,μ)=(0.0001,0.0001), α_k=0.001}	0.1	0.000191	0.000276
256 by 256	SGD NMF {D=3, (λ,μ)=(0.0001,0.0001), α_k=0.001}	0.1	4.648861e-05	5.775554e-05
512 by 512	BGD NMF{D=4, (λ,μ)=(0.0001,0.0001), α_k=0.001}	0.1	1.052556e-05	1.170430e-05
1024 by 1024	BGD NMF {D=4, (λ,μ)=(0.0001,0.0001), α_k=0.001}	0.1	1.600790e-06	1.695907e-06

Table 5.3: MF/NMF — QoS Minimum overhead required for $P_u = 10^{-1}W$

MIMO setup	Optimal hyperparameters	Min overhead	Train NMSE	Test NMSE
128 by 128	SGD MF {D=2, (λ,μ)=(0.0001,0.0001), α_k=1e-06}	0.1	0.115517	0.118776
256 by 256	BGD MF {D=3, (λ,μ)=(0.0001,0.0001), α_k=0.0001}	0.1	0.016475	0.016679
512 by 512	SGD NMF{D=4, (λ,μ)=(0.0001,0.0001), α_k=1e-06}	0.1	0.003371	0.003449
1024 by 1024	BGD MF {D=4, (λ,μ)=(0.0001,0.0001), α_k=1e-05}	0.1	0.001681	0.001948

Table 5.4: MF/NMF — 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, β_k=0.0001}	0.1	0.001144	0.002639
256 by 256	{(J=3, D=16), B=16, β_k=0.001}	0.1	3.941522e-05	3.948157e-05
512 by 512	{(J=3, D=64), B=32, β_k=0.0001}	0.1	3.305507e-05	3.335168e-05
1024 by 1024	{(J=3, D=64), B=64, β_k=0.0001}	0.1	9.810028e-06	9.857067e-06

Table 5.5: MLP — QoS Minimum overhead required for $P_u = 1W$

- Sub-figures 5.3c and 5.3b showcase the remarkable characteristic of the BCD-based models. These figures illustrate an immediate drop in loss values from the very initial iterations, signifying the efficiency of these models in quickly converging to optimal solutions.
- On the other hand, sub-figures 5.4a and 5.4b exhibit the progressive convergence of the cost function over the course of iterations when employing BGD-based models. These figures underscore the steady convergence behavior of these models, which may take longer to reach optimal solutions but do so effectively.

In summary, tables in (5.2) (5.3) (5.4) provides a comprehensive summary of the key findings, including the optimal signaling overhead ratio required for all proposed system configurations, the optimal model with the smallest total cost function, the corresponding combination of optimal hyperparameters, and the associated train/test error values. Remarkably, the MF models consistently maintain the same minimum signaling overhead requirement of 0.1, regardless of the transmitted power regime. This implies that these models can accurately predict with only 10% of sounded beams, resulting in a substantial 90% reduction in pilot signaling overhead compared to Exhaustive Beam Alignment, all while achieving negligible training and test errors.

5.5.2 MLP training and test QoS Performance

We define :

$$\text{Train NMSE} = \frac{1}{|\mathcal{K}|}\Big(\sum_{(u,i)\in\mathcal{K}}\Big(\frac{RSE_{u,i} - \sigma_J(\phi_{\mathbf{J}}...\sigma_2(\mathbf{\Phi_2}(\sigma_1(\phi_{\mathbf{1}}))))}{RSE_{u,i}}\Big)^2\Big) \tag{5.19}$$

Equivalently, the test NMSE is given by:

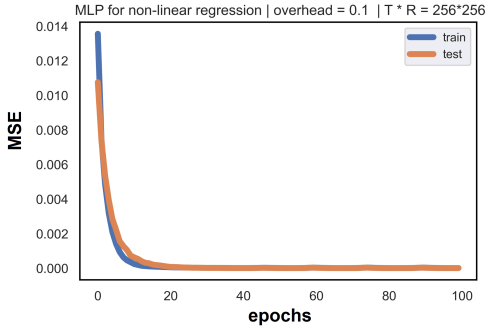
$$\text{Test NMSE} = \frac{1}{|\mathcal{L}|}\Big(\sum_{(u,i)\in\mathcal{L}}\Big(\frac{\widehat{RSE_{u,i}} - \sigma_J({\phi_{\mathbf{J}}}^*...\sigma_2({\mathbf{\Phi_2}}^*(\sigma_1({\phi_{\mathbf{1}}}^*))))}{\widehat{RSE_{u,i}}}\Big)^2\Big) \tag{5.20}$$

MIMO setup	Optimal hyperparameters	Min overhead	Train NMSE	Test NMSE
128 by 128	{(J=3, D=8), B=4, β_k=0.0001}	0.1	0.007569	0.007662
256 by 256	{(J=3, D=16), B=16, β_k=0.001}	0.1	0.000139	0.000288
512 by 512	{(J=3, D=64), B=32, β_k=0.0001}	0.1	5.419598e-05	5.756302e-05
1024 by 1024	{(J=3, D=64), B=64, β_k=0.0001}	0.1	1.184073e-05	1.72301e-05

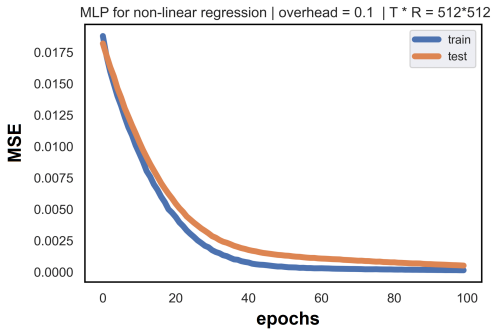
Table 5.6: MLP — QoS Minimum overhead required for $P_u = 10^{-1}W$

MIMO setup	Optimal hyperparameters	Min overhead	Train NMSE	Test NMSE
128 by 128	{(J=3, D=8), B=4, β_k=0.0001}	0.1	0.049559	0.071185
256 by 256	{(J=3, D=16), B=16, β_k=0.001}	0.1	0.017011	0.017634
512 by 512	{(J=3, D=64), B=32, β_k=0.0001}	0.1	0.000141	0.000666
1024 by 1024	{(J=3, D=64), B=64, β_k=0.0001}	0.1	1.700140e-04	1.702889e-04

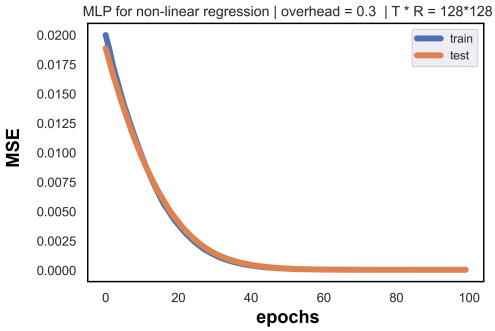
Table 5.7: MLP — QoS Minimum overhead required for $P_u = 10^{-2}W$



(a) Learning curve: 256×256 with overhead 0.1 MLP



(b) Learning curve: 512×512 with overhead 0.1 MLP



(c) Learning curve: 128×128 with overhead 0.3 MLP

Figure 5.5: MLP Learning curves

We used the same system configurations as for MF/NMF, resumed in (5.1). 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 the observations made for the Matrix Factorization models, we analyze the training performance of the MLP model by tracking the evolution of the cost function, represented by the Normalized Mean Squared Error, applied to the training samples in set $\mathcal{K}$ across iterations. This analysis reveals a consistent pattern characterized by a notably low range of error values and the coherent learning behavior of the MLP architecture. These results indicate that our shallow neural network effectively addresses the non-linear regression challenges inherent in our Beam Alignment process.

- Specifically, for massive MIMO setups, MLP achieves impressively low errors, reaching around 10^{-6} in scenarios with high transmitted power. However, it is worth noting that this cost value tends to increase as noise and interfer-

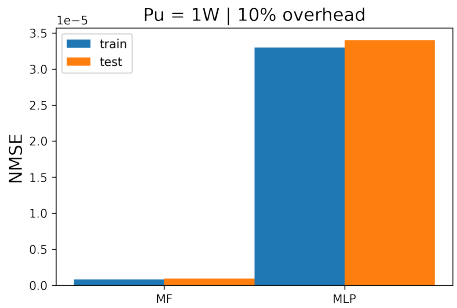
ence levels escalate. An interesting observation is that the training $NMSE$ decreases as we augment the size of the dataset matrix $\mathbf{S}$. This expansion provides MLP with more samples for training, facilitating enhanced feature extraction and improved prediction quality.

- Turning our attention to the evaluation of unknown beams, the test error values presented in the numerical results tables closely align with the training costs. This consistency suggests that MLP manages to maintain a balanced performance without signs of overfitting or underfitting, as corroborated by the learning curves. Furthermore, it's notable that the test loss experiences a similar sensitivity to the transmitted power regime as observed during the training process.

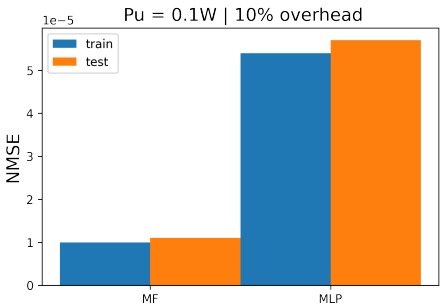
- Likewise, akin to the Gradient Descent based Matrix Factorization models, the learning curves of the Multi-Layer Perceptron depicted in Figure 5.5 exhibit a consistent pattern. These curves showcase a continuous, monotonic decrease in both training and test cost as the number of iterations progresses. This gradual convergence throughout the iterations culminates in training and test Normalized Mean Squared Error values reaching impressively low error levels at the final epoch. This convergence pattern underscores the capability of MLP to effectively address our specific problem and furnish a robust solution for Machine Learning-based Beam Alignment.

Transitioning to a Quality of Service perspective, we can extract valuable insights from tables in (5.5) (5.6) (5.7) , which summarizes the minimum (optimal) signaling overhead required to achieve successful beam-sounding while maintaining reliable prediction quality. Mirroring the observations made for MF/NMF models,MLP demonstrates consistency across various transmitted power scenarios. It consistently demands only 10% of the total beam-pairs to construct the RSE matrix effectively. This uniformity in signaling overhead requirements across different power levels underscores the robustness and efficiency of MLP in addressing the Beam Alignment task. MLP proves to be a versatile and effective solution, ensuring reliable performance without the need for excessive signaling overhead.

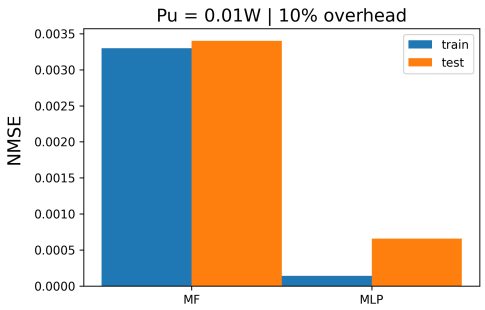
In summary, the performance of the MLP model in the training phase showcases its proficiency in resolving non-linear regression challenges associated with the Beam Alignment process. The model delivers low errors in massive MIMO setups under high transmitted power conditions. However, it is essential to recognize that the error increases in the presence of noise and interference. Additionally, the size of the dataset matrix has a positive impact on training performance. This robustness and consistency extend to the evaluation of unknown beams in the test phase, with the test error behavior closely mirroring that of the training process across varying transmitted power levels.



(a) 512×512 Train/Test $NMSE$ for $P_u = 1W$

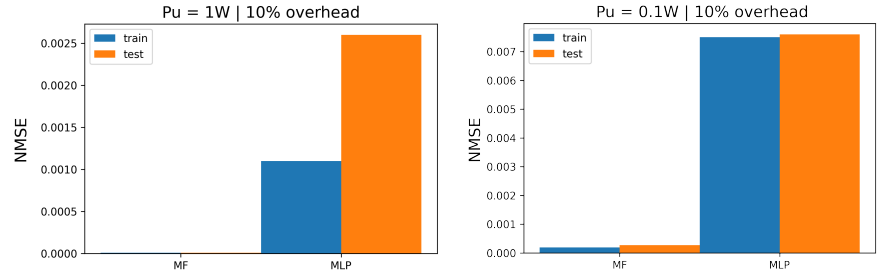


(b) 512×512 Train/Test $NMSE$ for $P_u = 10^{-1}W$



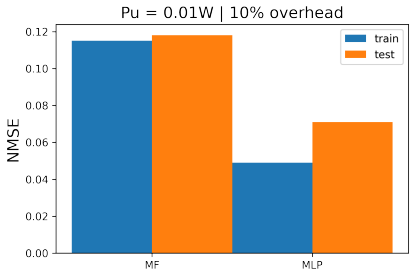
(c) 512×512 Train/Test $NMSE$ for $P_u = 10^{-2}W$

Figure 5.6: Train/Test $NMSE$ in function of P_u for MLP and MF for 512×512 using optimal overhead ratio



(a) 128×128 Train/Test $NMSE$ for $P_u = 1W$

(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 5.7: Train/Test $NMSE$ in function of P_u for MLP and MF for 128×128 using optimal overhead ratio

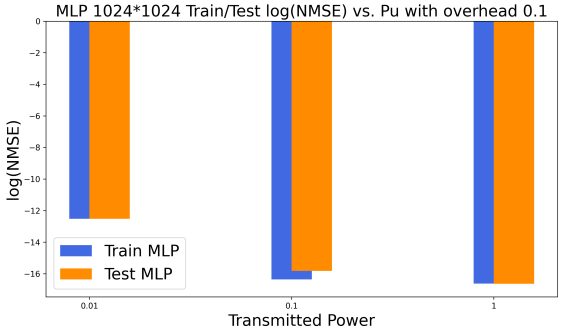
5.5.3 Comparative study of MF and MLP performances

In our analysis of the six different MF-based models, we've selected the most optimal one, characterized by the lowest test error, to represent the entire family of MF methods.

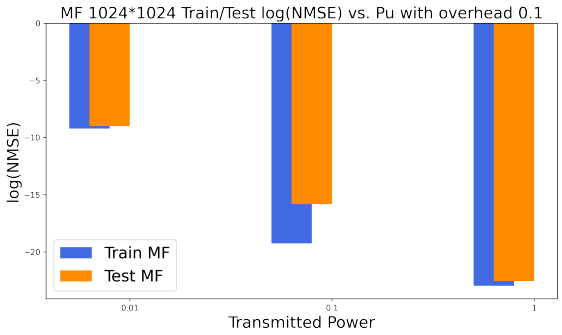
Upon scrutinizing the Quality of Service metrics presented in tables (5.2) (5.3) (5.4) for MF and (5.5) (5.6) (5.7) for MLP, we notice a clear influence of the transmitted power regime on prediction quality. Specifically, we observe a noticeable reduction in overall loss as we transition from higher-power scenarios to lower-power ones.

- For the MF and NMF models, this shift in power levels leads to a substantial increase in prediction error. In massive MIMO configurations such as 256, 512, and 1024, the losses are typically around 10^{-8}, but for smaller setups, they rise to approximately 10^{-4}. This highlights the sensitivity of MF and NMF models to changes in transmitted power, especially when moving to lower-power settings.

- When examining the MLP model, we notice a higher degree of resilience to variations in transmitted power. Although the overall loss does increase as transmitted power decreases, this impact is notably less pronounced compared to MF and NMF models. This robustness of MLP suggests that it



(a) MLP Train/Test $log(NMSE)$ in function of Pu using optimal overhead ratio



(b) MF Train/Test $log(NMSE)$ in function of P_u using optimal overhead ratio

Figure 5.8: $\log(NMSE)$ in function of P_u for 1024×1024 using optimal overhead ratio

can maintain reasonably consistent prediction quality even when operating in scenarios with varying transmitted power levels.

- Another notable empirical finding is that alterations in the values of P_u do not significantly affect the optimal hyperparameters selected through the cross-validation process. This stability in hyperparameter selection underscores the reliability and consistency of the chosen model configurations across different power scenarios.

- When tracking the evolution of training and test costs over the course of training iterations, we observe that all models exhibit balanced behavior. There are no evident signs of overfitting or underfitting. However, as transmitted power decreases, MF and NMF models tend to be more susceptible to increased train/test errors. In contrast, MLP maintains a more stable level of error, further emphasizing its robustness under changing power conditions.

In summary, the selection between MF/NMF models and MLP should not only consider prediction quality but also the models' sensitivity to variations in transmitted power. While MF/NMF models may excel in high-power contexts, they exhibit substantial performance degradation in low-power scenarios. On the other hand, MLP offers a more reliable and consistent performance profile, making it a suitable choice when the system needs to adapt to fluctuations in transmitted power levels. Moreover, the observed stability in hyperparameters simplifies model management across different power scenarios.

5.5.4 Similarities and Differences between models

All models required only 10% of the beams for training across all proposed massive setups. Furthermore, all proposed models are characterized by shallow neural architectures, with just a few hidden layers, to adhere to low-complexity constraints. Even among the largest configurations, the optimal model dimensions selected from cross-validation indicate the preference for small networks, eliminating the need for dense architectures.

Moreover, the MF-based models showcase exceptional accuracy, achieving impressively low loss values in the range of 10^{-8} for massive setups, particularly in the high P_u regime. Their cross-validation procedures reveal smaller grid searches with fewer hyperparameters to tune. However, it's worth noting that they tend to be slower when applied to high-dimensional MIMO setups.

In contrast, the MLP models strike an appealing balance between run-time complexity and prediction quality. They achieve loss values around 10^{-4} to 10^{-5} for massive configurations. Notably, the MLP exhibits remarkable robustness in the face of changes in P_u values.

- In Figure (5.6), we examine the performance of different models across various MIMO configurations and transmitted power levels. Starting with the 512×512 MIMO setup, subfigure (a) reveals that at a high power level

($P_u = 1W$), the Matrix Factorization model marginally outperforms the Multi-Layer Perceptron, with an $NMSE$ difference of approximately 10^{-1}. Moving to subfigure (b) ($P_u = 10^{-1}W$), MF maintains a slight performance edge, with an $NMSE$ difference of around 10^{-1}. However, in subfigure (c) ($P_u = 10^{-2}W$), we observe a shift in performance dynamics. Here, the MF model experiences a noticeable degradation in performance, resulting in an overall loss of approximately 10^{-3}, while MLP excels as the best performer. This transition suggests that MLP exhibits greater robustness in scenarios with low transmitted power, while MF and NMF tend to shine in high-power regimes.

- A similar pattern emerges when considering the 128×128 MIMO configuration in (5.7). Subfigure (a) highlights that in this setup, MF achieves significantly better performance compared to MLP, with an $NMSE$ difference of 10^{-4}. Subfigure (b) demonstrates the robustness of MLP, maintaining consistent error rates, while MF, despite experiencing a significant drop in performance (10^{-3}), still manages to outperform MLP. In subfigure (c), where P_u is at its lowest, MF performs at its worst across all simulations, while MLP consistently delivers the best prediction quality.

- On the other hand, MLP got slightly impacted with an overall loss of 10^{-1} and reaches the best quality of prediction. In Fig (5.8), we investigate the highest configuration 1024×1024. Similar conclusions for Fig (5.7) and Fig (5.6) hold for this figure in terms of best model (MF for $P_u = 1W$, $P_u = 10^{-1}W$ and MLP for $P_u = 10^{-2}W$).

In addition, we aim to investigate the overall impact of varying the transmitted power via logarithmic representation of the loss function. Thus, we track the $log(NMSE)$ values while switching from one P_u regime to another: In Fig (5.8), in subfigure (a), for MLP, the curve gap from low/medium is $log(NMSE)_{medium} - log(NMSE)_{low} \approx -16 - (-12) \approx -4$. The gap in the medium/high regimes is almost negligible ($log(NMSE)_{high} - log(NMSE)_{medium} \approx -16 - (-16) \approx 0.5$). Finally, in subfigure (c), the MF gap is around $log(NMSE)_{medium} - log(NMSE)_{low} \approx -17 - (-9) \approx -8$ and $log(NMSE)_{high} - log(NMSE)_{medium} \approx -22 - (-17) \approx -5$: 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.

5.6 Conclusion

This chapter marks the culmination of the second contribution to this thesis, where we harnessed the ability of both Matrix Factorization and Multi-Layer Perceptron to tackle the intricate challenge of partial and blind Beam Alignment. The chapter began by extending our system model and elevating the complexity of our experimental setup. Afterwards, we introduced the Multi Layer Perceptron architecture, input-output equations, problem statement and the formulated solution. Subsequently, we covered the numerical simulations for MF and for MLP and ended the chapter with a comparative study between both performances regarding the complexity/quality of prediction compromise. Again, 10% of total beam pairs are enough to accurately align the beams between UE and BS for both methods. However, the chapter stated the similarities and differences between models when varying the transmitted power. The outcomes of this comprehensive exploration equip us with valuable insights into the practical application of our methods in real-world scenarios, paving the way for more efficient and effective Beam Alignment in future communication systems. Thus, practical questions become mandatory for further industrial deployment. One particular problematic introduces the need of quantization before BA which radically changes the challenge framing and problem statement. This discussion is covered in details in the next chapter.

Chapter 6

Cascaded binary classifiers for Beam Alignment using 1-bit quantization

"I have this hope that there is a better way. Higher-level tools that actually let you see the structure of the software more clearly will be of tremendous value."

Guido van Rossum.

6.1 Introduction

This chapter marks the third contribution of the thesis, delving into a critical practical consideration: the quantization of beam pairs prior to beam alignment processing. In digital and hybrid beamforming architectures, digital systems operate on discrete values. By quantizing the received analog signals into digital values, the system can process the signals using digital signal processing techniques, making it easier to manipulate and analyze them. In the context of fully-analog architecture using ML units at BS, the need for quantizing the input-data helps in terms of signal representation, where Digital representation of signals allows easier storage, transmission, and manipulation of the data, before its ML processing. Quantization reduces the signal to a finite set of values, making it feasible to represent and transmit the information efficiently. Proper quantization ensures that the essential information is preserved while discarding irrelevant details, optimizing the use of available communication resources and offering compatibility with modern communication systems, protocols and infrastructure. This constraint fundamentally alters the problem formulation, rendering our Received Signal Energy dataset discrete due to the quantization process. From a Machine Learning perspective, this transition leads to a shift from non-linear regression, as explored in previous chapters, to the

domain of logistic regression. Despite this transformation, the core directions of our approach remain unchanged: we continue to operate within the realms of blind and partial Beam Alignment, where the objective is to complete a sparse dataset within the context of a "quantized" matrix. The ML model employed in this context is referred to as a classifier, tasked with discerning hidden patterns between the User Equipment and Base Station from a limited training set and making predictions for the test set. These predictions are now classifications, as the quantized dataset consists of classes or categories of beam pairs, depending on the chosen quantization scheme. From a Quality of Service perspective, this chapter scrutinizes both classification performance and complexity for the proposed cascaded structure of Binary Logistic Regression. This chapter seeks to address a dual question: What is the optimal quantization scheme, and correspondingly, the optimal overhead ratio required to ensure successful Beam Alignment and the establishment of a reliable initial link? The proposed approach empirically demonstrates the promising results, maintaining a 10% ratio of training samples and employing a binary quantization scheme to provide answers to these questions. The chapter includes the system model, the mathematical formulation of the problem with the quantization constraints, the input-output equations for the solution using our proposed learning approach, the experimental simulations and the performance evaluations.

6.2 System architecture

We consider a point-to-point mmWave Massive MIMO Uplink scenario, similar to the previous chapters, where a UE is equipped with N_T antennas linked to one RF chain and a BS is equipped with N_R antennas attached to N_{rf} RF chains, both wishing to align the optimal couple of beamformer/combiner which hold the maximum RSE. Evidently, the number of RF chains at both UE and BS is assumed to be much inferior to the number of antennas. The UE selects its fully-analog beamformer vector $\boldsymbol{f}_u \in \mathbb{C}^{N_T}$ from a DFT codebook holding the beam patterns, indexed as $u \in \mathcal{T}$, where the set $\mathcal{T}$, represents the codebook at UE. The BS similarly selects its fully-analog combiner matrix $\mathbf{W_i} \in \mathbb{C}^{N_R \times N_{rf}}$ from a DFT codebook $i \in \mathcal{R}$, where $\mathcal{R}$ represents the set of the codebook indexes at BS. $C_T = |\mathcal{T}|$, and $C_R = |\mathcal{R}|$ represent the cardinality of codebook at UE and BS resp. Thus, we denote *beam-pair* $(u, i) \in \mathcal{T} \times \mathcal{R}$, as beam $u \in \mathcal{T}$ from the codebook at UE, and beam $i \in \mathcal{R}$ from the codebook at BS. Besides, the received signal at BS, $\forall\ (u, i) \in \mathcal{T} \times \mathcal{R}$, is given by,

$$\mathbf{y_{u,i}} = \mathbf{W_i}^H \mathbf{G} \boldsymbol{f}_u s_u + \boldsymbol{n}_i,\ \forall\ (u, i) \in \mathcal{T} \times \mathcal{R} \tag{6.1}$$

where $\mathbf{G} \in \mathbb{C}^{\mathbb{N}_\mathbb{R} \times \mathbb{N}_\mathbb{T}}$ is the wideband mmWave MIMO channel (consistently assumed to be static and unknown to both UE and BS), $\boldsymbol{n}_i = \mathbf{W_i}^H \mathbf{n}$ is the unit-variance zero-mean AWGN and $s_u = \sqrt{P_u}$ where s_u, denotes the pilot symbol with transmitting power P_u corresponding to $\boldsymbol{f}_u$. Similar to the previous chapters, the received Signal to Noise ratio for the beam-pair (u, i) as:

$$\mathrm{SNR}_{u,i} = P_u ||\mathbf{W_i^H G} \boldsymbol{f}_u||_2^2, \ \forall \ (u,i) \in \mathcal{T} \times \mathcal{R} \tag{6.2}$$

Consistently, we approximate the Signal to Noise Ratio at the BS, with the instantaneous Received Signal Energy, i.e.,

$$\mathrm{RSE}_{u,i} = ||\mathbf{y_{u,i}}||_2^2, \ \forall (u,i) \in \mathcal{T} \times \mathcal{R}. \tag{6.3}$$

Benchmark: Recall the mathematical formulation of Brute Force BA:

$$(u^\star, i^\star) \quad = \quad \underset{(u,i) \in \mathcal{T} \times \mathcal{R}}{argmax} \quad \mathrm{RSE}_{u,i} \tag{6.4}$$

As a result, the *signaling overhead* is calculated as $\frac{|\mathcal{T}| \ \times \ |\mathcal{R}|}{N_{rf}} = \frac{C_T \times C_R}{N_{rf}}$, scaling as the product of both codebook sizes.

Partial Beam Sounding: The partial and blind BA is based on considering *sub-sampled and small-sized codebooks* of beams at UE and BS, $\mathcal{R}_\mathcal{S}$ and $\mathcal{T}_\mathcal{S}$ where, $\mathcal{R}_\mathcal{S} \subset \mathcal{R}$, $\mathcal{T}_\mathcal{S} \subset \mathcal{T}$, $|\mathcal{R}_\mathcal{S}| \ll |\mathcal{R}|$ and $|\mathcal{T}_\mathcal{S}| \ll |\mathcal{T}|$. The input-dataset matrix of our proposed approach can be equivalently expressed using the following incomplete RSE matrix, $\mathbf{S} \in \mathbb{R}^{C_T \times C_R} (:= \mathbb{R}^{|\mathcal{T}| \times |\mathcal{R}|})$ as,

$$[\ \mathbf{S}\ ]_{u,i} \ := \begin{cases} \mathrm{RSE}_{u,i} & , \text{ if } \ (u,i) \in \mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S} \\ \text{Unknown} & , \text{ if } \ (u,i) \notin \mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S} \end{cases} \tag{6.5}$$

where $[\mathbf{S}]_{u,i}$ denotes the coefficient (u,i) of $\mathbf{S}$, $\forall (u,i) \in \mathcal{T} \times \mathcal{R}$ The training set holding the sounded beams is denoted as $\mathcal{K} := \{(u,i) \mid (u,i) \in \mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S}\}$. The remaining unknown beams represent the test set, denoted as $\mathcal{V}$.

6.3 Binary Classification and one-bit Quantization

6.3.1 One-bit Quantization

After generating the $\mathbf{S}$ matrix given by the previous section, we propose the following quantization for the RSE values: let $Q_{u,i}$ denotes the non-uniform quantization function of $RSE_{u,i}$, $\forall (u,i) \in \mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S}$:

$$Q_{u,i} \in \mathbb{B} \ := \begin{cases} 0, \text{ if } \mathrm{RSE}_{u,i} < \beta \\ 1, \text{ if } \mathrm{RSE}_{u,i} \geq \beta, \ \forall (u,i) \in \mathcal{T}_\mathcal{S} \times \mathcal{R}_\mathcal{S} \end{cases} \tag{6.6}$$

For each binary class stated above, the number of training samples is equal for a balanced dataset, achieved by fixing the quantization threshold, β, as the median

value of the flattened and sorted $\mathbf{S}$ matrix.
Each training sample $(u,i) \in \mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}$ is characterized by a features-vector, $\mathbf{x}_{u,i} \in \mathbb{R}^d$ and the corresponding binary class/label $Q_{u,i} \in \mathbb{B}$. Thus, we define the training set, $\mathcal{K}$ as:

$$\mathcal{K} = \{(\mathbf{x}_{u,i} \in \mathbb{R}^d, Q_{u,i} \in \mathbb{B})\}_{\forall (u,i) \in \mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}} \tag{6.7}$$

In that sense, class 0 represents the bad quality beam pairs, which the model will iteratively eliminate, in contrast to good quality beams that we aim to keep, denoted as class 1.

6.3.2 Binary Logistic Regression

The input-output equations and the mathematical background of BLR are illustrated in this section for one (each) stage of the cascaded architecture. Given the vector of features, $\mathbf{x}_{u,i} \in \mathbb{R}^d$ for one training sample (u,i) and its related binary label, we denote $\mathcal{L}_{u,i}(\mathbf{w})$ as our individual loss for sample $(u,i) \in \mathcal{K}$, formulated as:

$$\mathcal{L}_{u,i}(\mathbf{w}) := (ln(2))^{-1} log_2(1 + exp^{-(\mathbf{w}^T \mathbf{x}_{u,i} Q_{u,i}))}), \tag{6.8}$$

$\forall (u,i) \in \mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}$ where $\mathbf{w} \in \mathbb{R}^d$ denote the Binary Logistic Regression weights. The empirical risk $\mathcal{L}(\mathbf{w})$, addressed as the average of the all individual losses of training, formulated as:

$$\mathcal{L}(\mathbf{w}) := \frac{1}{|\mathcal{K}|} \sum_{(u,i) \in \mathcal{K}} \mathcal{L}_{u,i}(\mathbf{w}) \tag{6.9}$$

The training phase aims to solve the *regularized* Empirical Risk Minimization, by finding $\mathbf{w}^\star$, the optimal weights from following optimization:

$$\mathbf{w}^* := f(\mathbf{w}) = \arg\min_{\mathbf{w} \in \mathbb{R}^d} \mathcal{L}(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2 =$$

$$\frac{1}{|\mathcal{K}| ln(2)} \sum_{(u,i) \in \mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}} log_2(1 + exp^{-(\mathbf{w}^T \mathbf{x}_{u,i} Q_{u,i}))}) + \lambda \|\mathbf{w}\|_2^2 \tag{6.10}$$

where $\lambda \|\mathbf{w}\|_2^2$ denotes the regularization term, used to combat overfitting symptoms and $f(\mathbf{w}) : \mathbb{R}^d \to \mathbb{R}$ represents the *regularized empirical risk* function. We propose to find the optimal BLR model $\mathbf{w}^\star$, using low-complexity gradient approach. The gradient for $f(\mathbf{w})$, $\nabla_{\mathbf{w}} f(\mathbf{w}) : \mathbb{R}^d \to \mathbb{R}^d$, is expressed as:

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = (ln(2)|\mathcal{K}|)^{-1} \sum_{(u,i) \in \mathcal{T}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}}} \left(\frac{-\mathbf{x}_{u,i} Q_{u,i} exp^{-(\mathbf{w}^T (\mathbf{x}_{u,i} Q_{u,i}))}}{1 + exp^{-(\mathbf{w}^T (\mathbf{x}_{u,i} Q_{u,i}))}}\right) + 2\lambda \mathbf{w} \tag{6.11}$$

The regularized empirical risk is strongly convex and the proof is found in Annex B of the manuscript. Thus, we have theoretical guarantees regarding the convergence

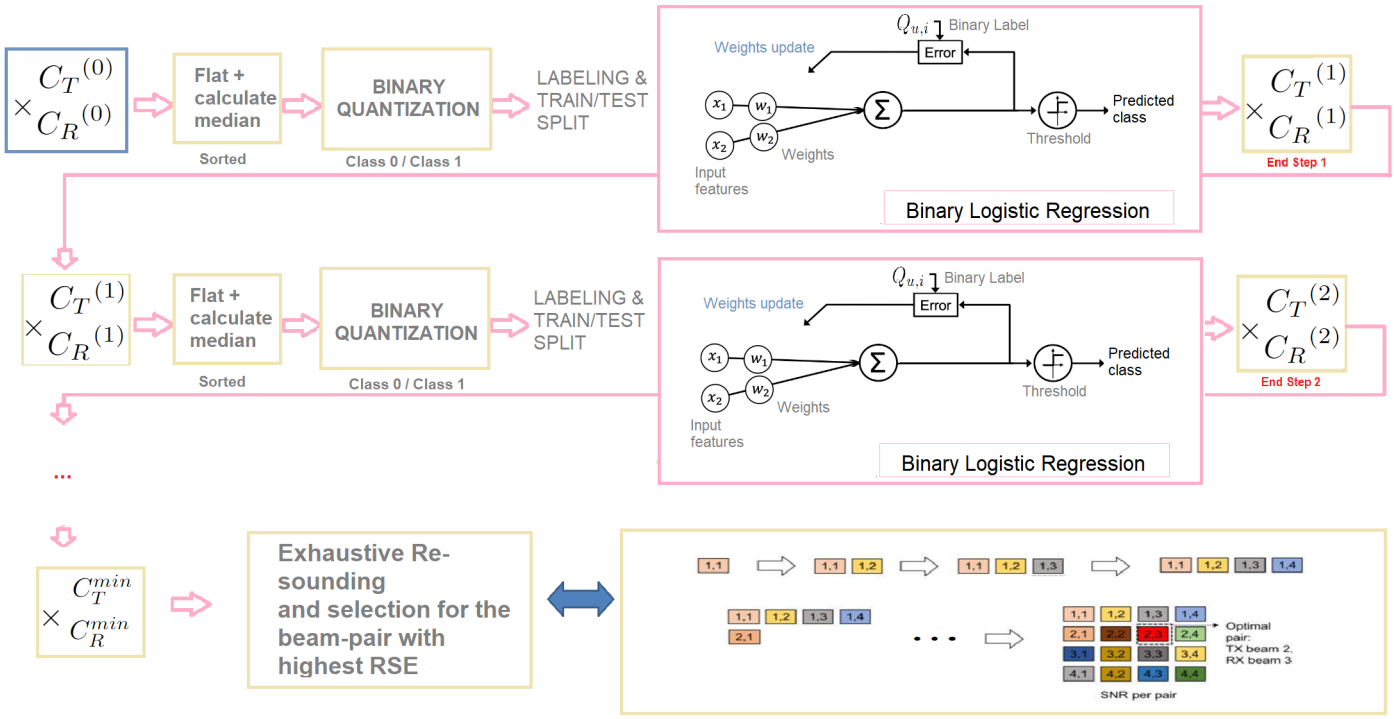


Figure 6.1: Cascaded binary logistic regression diagram representation

to optimal weights, where these weights are updated using Gradient Descent as following:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha_t \nabla_{\mathbf{w}} f(\mathbf{w}_t), \forall t \in \{1, .., N_{epochs}\} \tag{6.12}$$

where t is the index of the learning epoch, $\alpha_t = \alpha$ is the constant learning step size. Consequently, the BLR hyperparameters are the regularizer λ, the number of epochs N_{epochs}, the learning rate α, and the vector of features having dimension d. After solving the ERM problem, we obtain the optimal weights $\mathbf{w}^*$, used to predict the labels of unknown test samples:

$$\hat{Q_{u,i}} = \text{sign}(\mathbf{w}^{*T} \mathbf{x}_{u,i}), \quad \in \mathbb{B} \ \forall (u,i) \in \mathcal{V} \tag{6.13}$$

where $\hat{Q_{u,i}}$ is the predicted label for test sample $(u,i) \in \mathcal{V}$.

6.4 Proposed cascaded structure of Binary Logistic Regression

Our proposed approach includes a K-stage cascade, where a Binary Logistic Regression problem is solved within each cascade as illustrated in figure (6.1).

6.4.1 System architecture and input-output equations

The multi-level BLR cascaded structure includes K stage. In stage $k \in \{0, .., K-1\}$, we perform the steps below.

- 1) Let ${\mathcal{R}_{\mathcal{S}}}^{(k)} = BS$ codebook, ${\mathcal{T}_{\mathcal{S}}}^{(k)} = UE$ codebook at stage k, where $|{\mathcal{R}_{\mathcal{S}}}^{(k)}| = C_R^{(k)}$ and $|{\mathcal{T}_{\mathcal{S}}}^{(k)}| = C_T^{(k)}$. We sub-sample the codebooks from stage $k-1$, i.e., ${\mathcal{R}_{\mathcal{S}}}^{(k)} \subseteq {\mathcal{R}_{\mathcal{S}}}^{(k-1)}$ and ${\mathcal{T}_{\mathcal{S}}}^{(k)} \subseteq {\mathcal{T}_{\mathcal{S}}}^{(k-1)}, \forall k \in \{1, .., K-1\}$. We Choose ${\mathcal{T}_{\mathcal{S}}}^{(k)}, {\mathcal{R}_{\mathcal{S}}}^{(k)}$ such that $(|{\mathcal{T}_{\mathcal{S}}}^{(k-1)}|.|{\mathcal{R}_{\mathcal{S}}}^{(k-1)})|/(|{\mathcal{T}_{\mathcal{S}}}^{(k)}|.|{\mathcal{R}_{\mathcal{S}}}^{(k)})|$ meets the target η_k. Now, we sound the beam-pairs in the sub-sampled codebooks of stage k, $\forall\ (u^k, i^k) \in {\mathcal{T}_{\mathcal{S}}}^{(k)} \times {\mathcal{R}_{\mathcal{S}}}^{(k)}$. Afterwards, we build the RSE matrix $\mathbf{S}^{(k)} \in \mathbb{R}^{C_T^{(k)} \times C_R^{(k)}}$ as:

$$[\ \mathbf{S}\ ]_{u,i}^{(k)} := \begin{cases} \text{RSE}_{u,i}^{(k)} & , \text{ if } (u,i) \in \mathcal{T}_S^{(k)} \times \mathcal{R}_S^{(k)} \\ \text{Unknown}^{(k)} & , \text{ if } (u,i) \notin \mathcal{T}_S^{(k)} \times \mathcal{R}_S^{(k)} \end{cases} \tag{6.14}$$

- 2) Set threshold $\beta^{(k)}$ as the median of all entries in $\mathbf{S}^{(k)}$. Then, quantize all sounded entries in $\mathbf{S}^{(k)}$, $\forall (u,i) \in {\mathcal{T}_{\mathcal{S}}}^{(k)} \times {\mathcal{R}_{\mathcal{S}}}^{(k)}$:

$$Q_{u,i}^{(k)} \in \mathbb{B} := \begin{cases} 0, \text{ if } \text{RSE}_{u,i}^{(k)} < \beta^{(k)} \\ 1, \text{ if } \text{RSE}_{u,i}^{(k)} \geq \beta^{(k)}, \forall (u,i) \in {\mathcal{T}_{\mathcal{S}}}^{(k)} \times {\mathcal{R}_{\mathcal{S}}}^{(k)} \end{cases} \tag{6.15}$$

- 3) Select the features $(\mathbf{x}_{u,i}^{(k)})$ and labels $(Q_{u,i}^{(k)})$, for all training samples at stage k, $\mathcal{K}^{(k)}$, where:

$$\mathcal{K}^{(k)} = \{\mathbf{x}_{u,i}^{(k)} \in \mathbb{R}^d, Q_{u,i}^{(k)} \in \mathbb{B})\}_{\forall (u,i) \in {\mathcal{T}_{\mathcal{S}}}^{(k)} \times {\mathcal{R}_{\mathcal{S}}}^{(k)}} \tag{6.16}$$

- 4) Solve the empirical risk minimization in (6.17) at level k, $f(\mathbf{w}^{(k)})$, via Gradient Descent in (6.18):

$$\mathbf{w}^{*(k)} = \frac{1}{|\mathcal{K}^{(k)}| ln(2)} \sum_{(u,i) \in {\mathcal{T}_{\mathcal{S}}}^{(k)} \times {\mathcal{R}_{\mathcal{S}}}^{(k)}} log_2(1 + exp^{-({\mathbf{w}^T}^{(k)} \mathbf{x}_{u,i}^{(k)} Q_{u,i}^{(k)}))}) + \lambda^{(k)} \|\mathbf{w}^{(k)}\|_2^2 \tag{6.17}$$

$$\mathbf{w}_{t+1}^{(k)} = \mathbf{w}_t^{(k)} - \alpha_t^{(k)} \nabla_{\mathbf{w}^{(k)}}^{(k)} f(\mathbf{w}_t^{(k)}), \forall t \in \{1, .., N_{epochs}^{(k)}\} \tag{6.18}$$

where $\nabla_{\mathbf{w}^{(k)}}^{(k)} f(\mathbf{w}_t^{(k)})$ is obtained from (6.11) by substituting $\mathbf{w} = \mathbf{w}_t^{(k)}$, $\mathbf{x}_{u,i} = \mathbf{x}_{u,i}^{(k)}$, $Q_{u,i} = Q_{u,i}^{(k)}$, $\mathcal{T}_{\mathcal{S}} = {\mathcal{T}_{\mathcal{S}}}^{(k)}$, $\mathcal{R}_{\mathcal{S}} = {\mathcal{R}_{\mathcal{S}}}^{(k)}$, $\lambda = \lambda^{(k)}$

- 5) After the Gradient Descent iterations in (6.18) converge, we get the ideal BLR model for stage k, $\mathbf{w}^{*(k)}$. Thus, we predict the missing unknown labels, the indexes related to the test set $\mathcal{V}^{(k)}$. At stage k, the predicted label $\hat{Q_{u,i}^{(k)}}$ for pair $(u,i) \in \mathcal{V}^{(k)}$ is:

$$\hat{Q_{u,i}^{(k)}} = \text{sign}({\mathbf{w}^\star}^{(k)} \mathbf{x}_{u,i}^{(k)}) \forall (u,i) \in \mathcal{V}^{(k)} \tag{6.19}$$

Afterwards, we eliminate the beam pair indexes of class 0 beams and store the beam pair indexes of class 1.

The final stage $k = K$, is enabled when dimensions of BS and UE codebooks reach pre-determined value, which means when $|\mathcal{T_S}^{(K)}| = C_T^{min}$, $|\mathcal{R_S}^{(K)}| = C_R^{min}$. Subsequently, the Brute Force BA sounds all the beam-pairs in $\mathcal{T_S}^{(K)}$ and $\mathcal{R_S}^{(K)}$, i.e., $\forall(u,i) \in \mathcal{T_S}^{(K)} \times \mathcal{R_S}^{(K)}$. The resulting $C_R^{min} \times C_T^{min}$ RSE matrix is searched to find optimal beam-pair $(u^\star, i^\star)$, with the largest RSE. No learning nor quantization is required at stage K.

6.4.2 Analysis: algorithm convergence, computational complexity, signaling overhead

The principle motivation behind the use of BLR is its remarkably low complexity, which states a fundamental condition for BA in mmWave massive MIMO. Based on GD, the *computational complexity* ζ_k in stage $k \in \{0, .., K-1\}$ is given by:

$$\zeta_k = \mathcal{O}(\mathcal{T_S}^{(k)} \times \mathcal{R_S}^{(k)}) \tag{6.20}$$

We formulate the ratio of beam couples to be deleted at stage k , $\delta_k \forall k \in \{1, ..., K\}$, as the fraction of the output to input dimensions:

$$\{\delta_k\}_{k=1}^{K} = \frac{{C_T}^{(k)} \times {C_R}^{(k)}}{{C_T}^{(k-1)} \times {C_R}^{(k-1)}} \tag{6.21}$$

We then introduce $\eta_k, \forall k \in \{0, ..., K-1\}$, as the ratio between the number of training samples and the total number of samples in stage k:

$$\{\eta_k\}_{k=0}^{K-1} = \frac{\mathcal{T_S}^{(k)} \times \mathcal{R_S}^{(k)}}{{C_T}^{(k)} \times {C_R}^{(k)}} \tag{6.22}$$

The *aggregated* signaling overhead, denoted η, sums the overheads in all stages:

$$\begin{aligned} \eta = (C_T C_R)\eta_0 + (C_T C_R)\delta_1\eta_0 + (C_T C_R)\delta_1\delta_2\eta_1 + ... \\ + (C_T C_R)\delta_1\delta_2..\delta_K\eta_{K-1} + C_T^{min} C_R^{min} \end{aligned} \tag{6.23}$$

The *total* signaling overhead ratio, denoted γ, is the fraction of the overhead of proposed approach by the overhead of Brute Force BA:

$$\gamma = \frac{\eta}{{C_T}^{(0)} \times {C_R}^{(0)}} \tag{6.24}$$

6.4.3 Cascaded-BLR based BA Algorithm

Our proposed approach is detailed in Algorithm (7).

6.5 Numerical Simulations

In the experimental setup, we first propose to set the number of antennas as equal to the sizes of the related codebooks. The number of stages of the proposed cascade

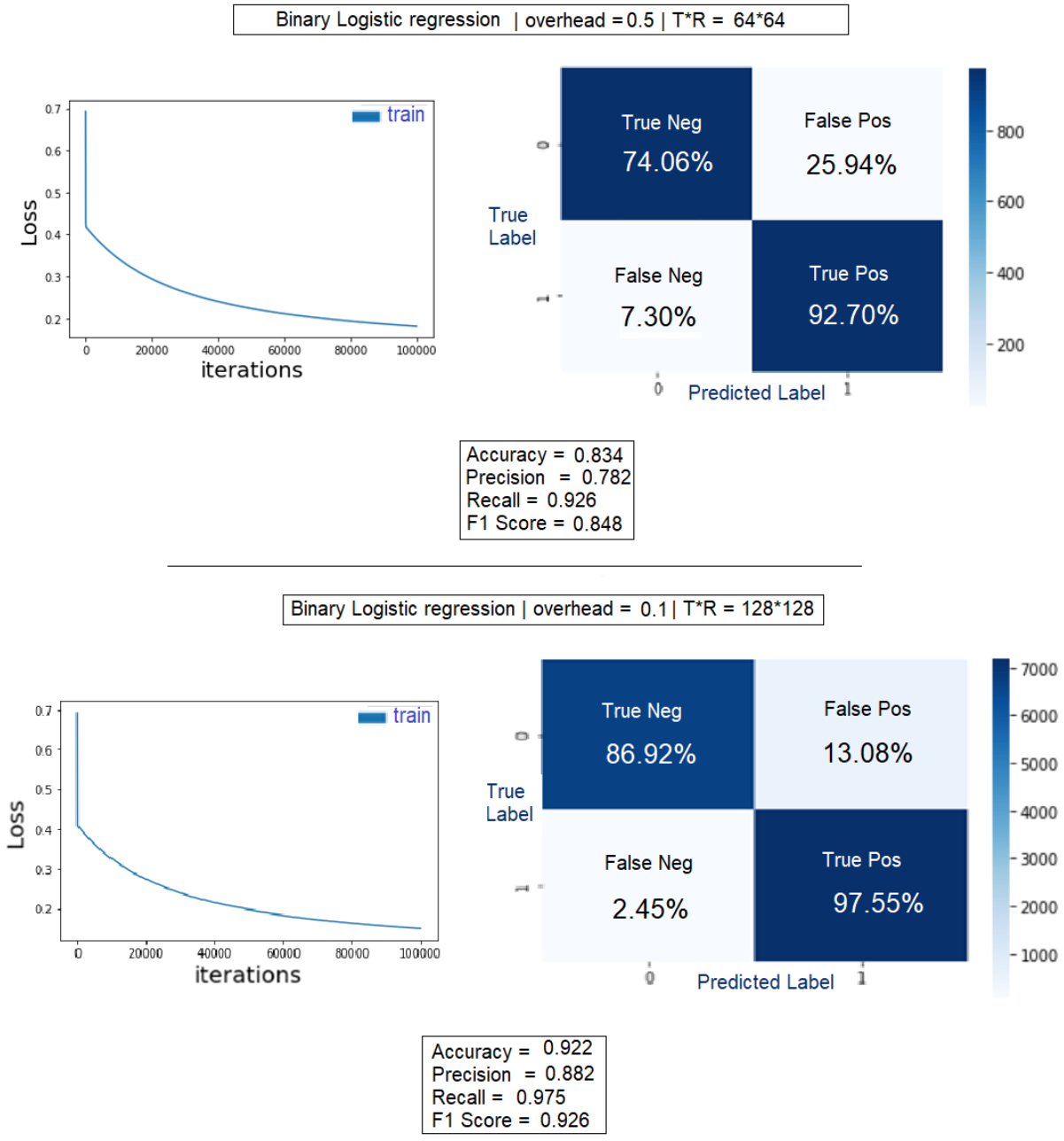


Figure 6.2: Models performance evaluation for 64×64 and 128×128: learning curves, confusion matrix, accuracy, precision, recall and F1-score

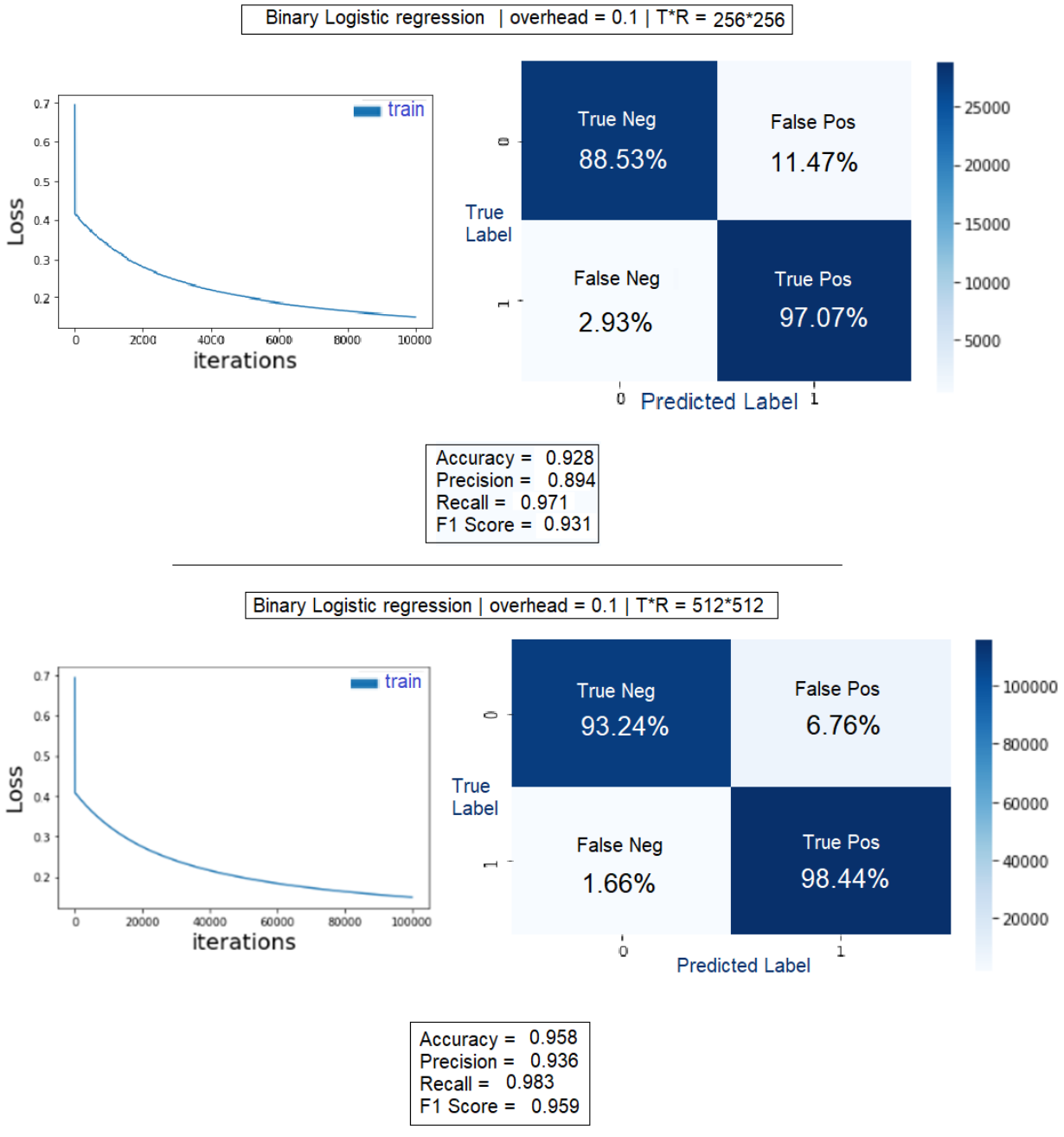


Figure 6.3: Models performance evaluation for 256×256 and 512×512 : learning curves, confusion matrix, accuracy, precision, recall and F1-score

Algorithm 7 Proposed BA using cascaded-BLR

Input: $\{\mathbf{f}_u\}_{\forall u \in \mathcal{T}}$, $\{\mathbf{W}_i\}_{\forall i \in \mathcal{R}}$, η_k, δ_k, (C_T^{min}, C_R^{min})

1- initialize δ_k, η_k and $\mathbf{w}$ and randomly generate codebooks at UE and BS, ${\mathcal{T}_{\mathcal{S}}}^{(k)}, {\mathcal{R}_{\mathcal{S}}}^{(k)}$ w.r.t η_k.

2- sound beam-pairs from training set, $\mathcal{K}^{(k)} := {\mathcal{T}_{\mathcal{S}}}^{(k)} \times {\mathcal{R}_{\mathcal{S}}}^{(k)}$.

3- store sounded $RSE_{u,i}^{(k)}$ values, in (6.3), so that we generate $RSE_{u,i}^{(k)}$ input dataset matrix $\mathbf{S}^{(k)}$, in (6.14).

4- flatten $\mathbf{S}^{(k)}$ into a sorted vector and fix the quantization threshold $\beta^{(k)}$ as the median, as in (6.15)

5- attribute to each beam-pair its corresponding class and split features $(\mathbf{x}_{u,i}^{(k)})$ and labels $(Q_{u,i}^{(k)})$, as in (6.16)

6- BLR processing:
solve the ERM in (6.17) where weighs update follows the equation (6.18).
at $N_{epochs}^{(k)}$, return optimal weights vector $\mathbf{w}^{*(k)}$.

7- use $\mathbf{w}^{*(k)}$ to predict the classes of the test set unknown beams $\mathcal{V}^{(k)}$ in (6.14).

8- delete the class-0 beams and get the new input dataset matrix holding class-1 beams only.

9- $k = k+1$: repeat instructions from step 2 to step 8 w.r.t η_k and δ_k until the dimensions of the final-output matrix reach $C_T^{min} \times C_R^{min}$ holding best beam-pairs.

10- exhaustively sound all best beam-pairs and select the couple with the highest RSE.

Output: $\boldsymbol{f}_{u^\star}$, $\mathbf{W}_{i^\star}$

is $K = 5$. ${C_T}^{(0)} = 1024$, ${C_R}^{(0)} = 1024$, $C_T^{min} = 32$, $C_R^{min} = 32$. Besides, we fix $\delta_k = 0.25, \forall k$. The transmit symbol power is $P_u = 1$, the number of $OFDM$ sub-carriers is 64 and the number of paths is 1, simulating a LoS scenario. We use DFT codebooks at UE and BS and we transmit one symbol. We set the overhead ratios for the different stages as, $\eta_1 = \eta_2 = \eta_3 = \eta_4 = 0.1$ and $\eta_5 = 0.5$. In addition, for stage k, we use the following feature, $\mathbf{x}_{u,i}^{(k)} = [\ u\ , i\ ]^T, \forall (u,i) \in {\mathcal{T}_{\mathcal{S}}}^{(k)} \times {\mathcal{R}_{\mathcal{S}}}^{(k)}$, and the related model dimension is $d = 2$, for all results. We propose an off-line grid-search cross-validation in order to obtain the optimal hyperparameters $\{(\alpha^{(k)}, \lambda^{(k)}, N_{epoch}^{(k)})\}_{k=1}^{K}$.

We introduce the following metrics for binary classification:

- True Positive (TP): if true label $Q_{u,i} = 1$, and predicted label $\hat{Q_{u,i}} = 1$
- True Negative (TN): if true label $Q_{u,i} = 0$, and predicted label $\hat{Q_{u,i}} = 0$
- False Positive (FP): if true label $Q_{u,i} = 1$, and predicted label $\hat{Q_{u,i}} = 0$
- False Negative (FN): if true label $Q_{u,i} = 0$, and predicted label $\hat{Q_{u,i}} = 1$.

These evaluation metrics are usually included into a confusion matrix: $\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$.

In addition, we define the following metrics of evaluation for Binary Classifcation:

- The *accuracy* is given by:

$$Accu = \frac{TP + TN}{TP + FP + TN + FN} \tag{6.25}$$

- The *precision*, also called Positive Predictive Value, is expressed as:

$$Prec = \frac{TP}{TP + FP} \tag{6.26}$$

- The *recall*, also denoted as sensitivity in binary classification, is formulated as:

$$Rec = \frac{TP}{TP + FN} \tag{6.27}$$

- The *F1 score* is given by:

$$F_1 = \frac{2 \times precision \times recall}{pecision + recall} = \frac{2TP}{2TP + FP + FN} \tag{6.28}$$

6.5.1 Train/Test Performance

Concerning the train performance, in figures (6.2) (6.3), the cost in function of the iterations for training samples, monotonically converge to low values with no overfitting nor underfitting symptoms for all proposed input dimensions across the cascade. Test performance is based on evaluating how precise was the BLR in predicting the class labels for unknown beam pairs using confusion matrices, accuracies, precisions, recalls and F1 scores. Thus, for all proposed input setups, BLR reached high classification scores with non-sounded beams: for input-matrix configurations from 64×64 to 512×512, the confusion matrices are diagonal where the predicted binary labels accurately matches the true binary labels for each class. Even through the smallest configurations and relying on small amount of training samples, BLR kept good classification scores and accurate predictions. These scores get extremely closer to one, illustrating the optimal score, as long as the input dimensions are large 'enough': in that case, the training set is considerably sufficient, and the prediction error may be low.

6.5.2 Total signaling overhead ratio

Given $\eta_1 = \eta_2 = \eta_3 = \eta_4 = 0.1, \eta_5 = 0.5$ and $\delta = 0.25$, the aggregated overhead of the cascaded structure of BLR, stated in (6.23), is equal to: $\eta = 1024 \times 1024 \times 0.1 + (1024 \times 1024 \times 0.25) \times 0.1 + (1024 \times 1024 \times 0.25^2) \times 0.1 + (1024 \times 1024 \times 0.25^3) \times 0.1 + (1024 \times 1024 \times 0.25^4) \times 0.5 + 32 \times 32 \times 1 = 142336$. Subsequently, the total overhead in (6.24) is equal to $\gamma = \frac{142336}{1024 \times 1024} = 0.1357$ which states a promising solution regarding the Beam Alignment of mmWave massive MIMO systems with one-bit quantization scheme and low total signaling overhead ratio.

Remark: from ML perspective, we implemented the proposed cascaded structure using a Multi Layer Perceptron MLP instead of BLR at each stage, so that we investigate the trade-off of complexity and quality of prediction. The error function of the proposed feed forward architecture is the Binary Cross Entropy Loss. The computational complexity of the MLP is orders of magnitude larger than that of the proposed Binary Logistic Regression. Besides, tuning the hyperparameters takes substantially more time for the neural network. We empirically observed that MLP performance doesn't exceed 6% of overall gain in best case scenario. It's noteworthy to state that from Wireless Communication perspective, a cascaded of Multi Layer Perceptrons is not applicable in the proposed system architecture given the low complexity constraints (1-bit quantization, 1 RF chain).

6.6 Conclusion

This chapter has unveiled the third contribution of this thesis, focusing on the ramifications of quantization within our proposed Beam Alignment methods. The chapter commenced with the establishment of the system architecture and model equations. Subsequently, it ventured into the mathematical formulation of quantization constraints, laying the groundwork for the creation of the corresponding quantized Received Signal Energy dataset. The learning approach was then elucidated, delineating the cascade of Binary Logistic Regression layers and their respective input-output equations at each stage. Ultimately, the numerical findings have indicated that our consistent result regarding the optimal overhead ratio (10% throughout the entirety of this thesis) can be harmonized with an assertive one-bit binary quantization. Remarkably, this adjustment doesn't compromise the predictive quality while adhering to stringent low-complexity prerequisites. These outcomes and inferences from this chapter set the stage for three major research directions: scalability to multi-user scenario, adding robustness factors in the system equations and interpretability as we need to open the black box of the proposed neural architectures and aim to explain their predictions. These conclusions and perspectives are expounded upon in the last chapter of this manuscript while the generalization of our proposed approach to the Multi User scenario is presented in the next chapter.

Chapter 7

Convolutional Neural Network and Auto Encoder for Multi User Beam Management in mmWave massive MIMO

"The past resembles the future more than one drop of water resembles another."

Ibn Khaldoun.

7.1 Introduction

In this chapter, we embark on an exploration of the scalability of our proposed methods within a multi-user Uplink system configuration. Recognizing the challenges and constraints of a MU-MIMO blind approach, we opt for a new strategy. Instead of relying on blind techniques, we choose to sound the channel before initiating the Beam Alignment process based on CSI. This provides us with precise values of the signal to interference and noise ratio for each beam-pair between User Equipments and the Base Station, both equipped with Discrete Fourier Transform codebooks containing equispaced beams.

To accommodate this generalization towards a reliable multi-user system, we extend our system architecture and propose two methodologies:

- Process each user separately: in this approach, we dedicate one Machine Learning processing unit for each user, complete with all the necessary hardware and software resources. This allows us to simultaneously process the Beam Alignment task for K users, each using its own set of shallow and low-complexity ML models and requirements. The perspective here is to optimize and evolve these models towards a distributed learning setup, where these learning units can exchange data, continually train, and optimize their weights based on all

users activities, aiming to enhance the overall system efficiency. The ML models we investigate for this use case include a shallow symmetric auto-encoder and a shallow feed-forward architecture.

- Process all users at once: in contrast, this approach involves a single ML processing unit at the BS. This unit is equipped with the necessary hardware requirements to accommodate a more extensive and sophisticated neural network. We switch from simple Signal-to-Interference-plus-Noise Ratio matrices for each user to a unified high dimensional tensor that combines all these matrices into a single set of training and test samples. The Partial Beam Alignment task is then formulated as a tensor completion procedure. This approach lends itself to the use of Convolutional Neural Network.

The multi-user scenario considered in this manuscript provides insights into the strengths and limitations of these proposed approaches. Primary simulations suggest that overall, models demonstrate efficient and reliable performance but may request further refinement to meet robustness requirements. Consequently, this chapter sets the stage for future research directions and perspectives, potentially leading to significant contributions and industrial applications.

7.2 SotA Multi-user Beam Alignment

Regarding the particularity of multi user BA, we propose to re-introduce the corresponding SotA approaches, including additional references, aiming to specialize the general literature overview we presented in chapter 3, to the specific multi user Beam Management. This literature survey introduces two families of BA models, similarly to the single user. Authors in [65] presented Exhaustive BA applied on WLAN/WPAN scenario. The approach is simple and effective but exhibit high complexity. The Exhaustive BA for Cellular networks is proposed in [66]. Hierarchical Search in multi user cellular systems is investigated in [67] and [68], proposing to reduce the number of beams to be sounded. However, the approach may undergo severe misalignment error propagation. Still in the context of Cellular networks, authors in [69] reduced the number of beams via two-stage search but increased power consumption. Same remark holds for [70] where Compressed Sensing is used. One limitation of this approach is the absence of antenna gain during the measurement step. Side Information based BA is illustrated in [71] for High-Speed-Train Communications, for Vehicle-to-Everything in [72] and for Unnamed-Aerial-Vehicles in [73], aiming to limit the search space in limited areas but mainly need additional sensing equipments. Even when they reduce the pilot overhead, the implementation of these classical approaches in massive MIMO multi user scenarios is a challenge, and more reduction on the training samples is required to guarantee their efficiency. The robustness of these methods in cases where users are not static, is another serious research direction for traditional BA, in reference to Beam Tracking methods, mentioned in Chapter 3.

On the other hand, Supervised Learning is investigated in [74] in the context of Vehicle-to-Everything communications and in Cellular networks in [75]. These methods considerably reduced the overhead with wide range of applications and high accuracy but with expensive cost to collect training samples. Reinforcement Learning or multi user Cellular BA in [76] exhibit negligible overhead but lacks the capability of making complex decision using existing domain knowledge. In [77], authors introduced a Bidirectional Recurrent Neural Network and Short-Long-Term-Memory network to overcome the time-latency, but with an energy consuming digital architecture. Same approach is presented in [78], using a recurrent neural network based on users orientation information and a reference signal received power. Its limitation resides in the continuous need for more simulation data and possibly field measurement data. In [79], authors discuss a Cell-free distributed MIMO approach, jointly handling analog Beam Selection and Digital Beam forming using the extended Saleh-Valenzuela geometric channel, generating the SINR values for each user, similarly to our approach, in the next section. In addition to the remarkable sum-rate losses achieved by the proposed Supervised Learning models, selectively shutting of some RF chains helps save significant power consumption. In [80], authors handle the multi user BA using CNN and SVD-based beams with limited CSI feedback, applied on Quadriga dataset. Hybrid Precoding, Tensor Dictionary Manifold and Supervised Learning tools are investigated in [81] [82] [83] for channel estimation and coverage optimization. Finally, the new trend in multi user Beam Management is related to the promising results observed when using Reflective Intelligent Surfaces, such as in [84] and [85], with low pilot overhead and high alignment accuracy. However, their implementation is challenging and exhibits higher complexity than other SotA methods.

To sum-up, we propose to encounter the complexity-limitations of SotA approaches with fully-analog low-complexity system-architecture, aiming to reach low pilot overhead ratios and high accuracy matrix/tensor completion using shallow neural networks.

7.3 Multi-user system model

We consider an Uplink mmWave MU-MIMO setup with one BS and K users. These UEs are equipped with N_T antennas attached to one RF chain for each user separately, while the BS is equipped with N_R massive antennas linked to N_{rf} RF chains. Each user transmits one symbol from the symbols vector ($\mathbf{s} \in \mathbb{C}^K$) and gets the same value of transmitted power (P_u). The architecture is still low-complex and fully-analog but requires the CSI-based channel estimation before BA. In addition, the technical objective doesn't change: we aim to accurately align the optimal precoder/equalizer for all users where each user gets the ideal pair of beamformer/combiner holding the maximum SINR. As we did in previous chapters, each UE selects its analog beamformer $\mathbf{F}_u \in \mathbb{C}^{N_T \times N_T}$ from a DFT codebook for each user, containing beam patterns, indexed as $u \in \mathcal{T}$. The precoding matrix contains K beamformer vectors corresponding to each user, denoted $\mathbf{f}_u \in \mathbb{C}^{N_T}$. On the other

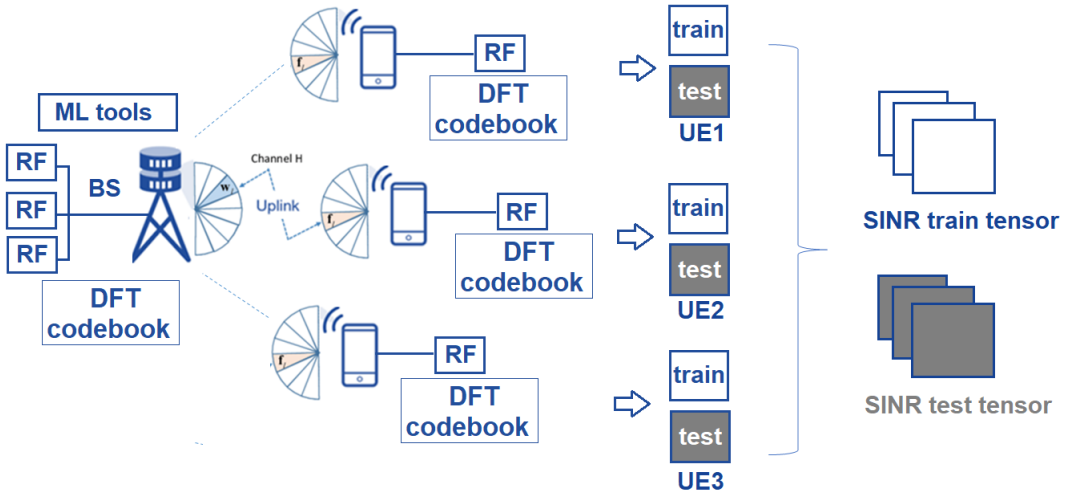


Figure 7.1: Simplified diagram representation of the proposed Uplink multi-user architecture with 3 UEs

side of the transmission, the BS selects its analog combiner matrix $\mathbf{W}_i^H \in \mathbb{C}^{N_R \times N_{rf}}$ from a DFT codebook $i \in \mathcal{R}$. The proposed system architecture is simplified and resumed in figure (7.1)

The signaling overhead Ω is defined as the total number of pilots needed for BA, i.e., the total number of samples multiplied by the signaling overhead ratio η. We denote T as the total number of time slots in the frame in order to recall the effective rate for user k, $r^{(k)}$, expressed as:

$$r^{(k)} = (1 - \frac{\Omega}{T}) \log_2(1 + \text{SINR}^{(k)}) \tag{7.1}$$

Thus, we aim to generate the SINR value for each beam couple for each user. Therefore, we first represent beam-pair $(u, i) \in \mathcal{T} \times \mathcal{R}$. Consequently, the total received signal at BS is given by:

$$\mathbf{Y} = \sum_{k=1}^{K} \mathbf{Y}_k = \sum_{k=1}^{K} \mathbf{W}^H \mathbf{G} \mathbf{f}_k s_k + \mathbf{n} \tag{7.2}$$

where $\mathbf{G} \in \mathbb{C}^{N_R \times N_T} = \sqrt{\frac{1}{N_c}} \sum_{k=1}^{K} \sum_{l=1}^{N_c} \mathbf{H}_{l,k}$ follows the same geometric Saleh-Valenzuela model [22], oriented to the multi user case as in [41], with N_c the number of sub-paths per user. We denote $\mathbf{n_i} = \mathbf{W_i}^H \mathbf{n}$ as the zero-mean unit-variance AWGN.

7.4 SINR tensor dataset: problem formulation

The dataset generated by the proposed model architecture is based on the SINR values for each beam pair (u,i), for each user (k), as in [79]:

$$SINR_{u,i}^{(k)} = \frac{P_u \|\mathbf{W_i}^H \mathbf{G} \mathbf{f}_u^{(k)} s_k\|_2^2}{P_u \sum_{j=1, j\neq k}^{K} \|\mathbf{W_i}^H \mathbf{G} \mathbf{f}_u^{(j)} s_j\|_2^2 + \mathbf{n_i}}, \forall k \in \{1, .., K\} \tag{7.3}$$

Consistently, our benchmark is the Exhaustive BA, and all traditional methods based on it [2] [3]. The selection of the best beam is the result of the highest SINR from a total sounding for all possible pairs regarding codebooks at UEs and BS. For each user k, the optimal codebooks indexes are formulated as:

$$(u^\star, i^\star)^{(k)} = \underset{(u,i)\in\mathcal{T}\times\mathcal{R}}{argmax} \quad \text{SINR}_{u,i}^{(k)}, \forall k \in \{1, .., K\} \tag{7.4}$$

The SINR tensor, $\mathbf{S} \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{R}|\times K}$) is a stack of incomplete SINR matrices for each user. Thus, $\forall k \in \{1, .., K\}$, each SINR matrix is equivalently formulated as:

$$[\ \mathbf{S}\ ]_{u,i}^{(k)} := \begin{cases} \text{SINR}_{u,i}^{(k)} & , \text{ if } (u,i) \in {\mathcal{T}_\mathcal{S}}^{(k)} \times {\mathcal{R}_\mathcal{S}}^{(k)} \\ \text{Unknown} & , \text{ if } (u,i) \notin {\mathcal{T}_\mathcal{S}}^{(k)} \times {\mathcal{R}_\mathcal{S}}^{(k)} \end{cases} \tag{7.5}$$

As we did before, ${\mathcal{T}_\mathcal{S}}^{(k)}$ and ${\mathcal{R}_\mathcal{S}}^{(k)}$ are the cardinalities of the sub-sampled codebooks for each user k in order to accomplish the multi user partial BA task. Similarly to the previous chapters, the training set holding sounded SINR values is denoted $\mathcal{K}$ and the test set containing the unknowns is denoted $\mathcal{L}$.

On the other hand, the overhead ratio per user, denoted η_k, is the fraction of the pilot overhead of our proposed approach for user k divided by the overhead of Exhaustive BA:

$$\eta_k := \frac{\text{overhead of ML based BA}}{\text{overhead of exhaustive BA}} = \frac{|{\mathcal{T}_\mathcal{S}}^{(k)}| \times |{\mathcal{R}_\mathcal{S}}^{(k)}|}{|\mathcal{T}| \times |\mathcal{R}|}, \forall k \in \{1, .., K\} \tag{7.6}$$

The sum of all signaling overhead ratios introduce the Total overhead of the proposed solution, η, applicable to both approaches (generalized point-to-point and tensor completion):

$$\eta = \frac{1}{K} \sum_{k=1}^{K} \eta_k \tag{7.7}$$

7.5 Proposed solutions using AE, MLP and CNN

A simplified illustration of the proposed solution is proposed in figure (7.2). In the proposed Multi-User Beam Alignment algorithm, the system begins by generating random codebooks at both the user equipments and base station antennas,

ensuring adherence to the specified oversampling ratio. Beam pairs are then systematically probed from the training set for each user, and the corresponding Signal-to-Interference-plus-Noise Ratio values are stored, forming a SINR dataset tensor.

To align the beams effectively, the algorithm offers two distinct approaches. The first approach, denoted as Generalized point-to-point or Generalized matrix completion, involves dividing the SINR tensor into incomplete matrices. Shallow Multi-Layer Perceptrons and Symmetric Auto-Encoders are fine-tuned specifically for each user, enabling the completion of the SINR matrices. The second approach, defined as Tensor Completion, employs a more complex architecture, Convolutional Neural Network, fine-tuned across all users to complete the SINR tensor. The algorithm then evaluates and selects the optimal codebook indexes based on the highest SINR values obtained, providing the optimal transmit beamforming vector for the users ($\mathbf{F}_{u^\star}$) and the best receive combining vector for the base station antennas ($\mathbf{W}_{i^\star}$).

Besides, the architectural limitations of the first approach are explained by the need of one ML processing unit for each user while the second approach requires one unit for all of them. However, shallow Feed Forward models and shallow Auto-Encoders are less complex/greedy in terms of computational resources than the CNN. This compromise is empirically investigated in the numerical simulations.

Finally, both approaches are neatly tied together and presented in Algorithm (8)

Algorithm 8 Proposed Multi User Beam Alignment

Input: $\{\mathbf{F}_u\}_{\forall u \in \mathcal{T}}$, $\{\mathbf{W}_i\}_{\forall i \in \mathcal{R}}$, η, η_k, K

[1-] Generate randomly codebooks at UEs and BS, $\mathcal{T}_{\mathcal{S}}{}^{(k)}, \mathcal{R}_{\mathcal{S}}{}^{(k)}$, satisfying $(|\mathcal{T}_{\mathcal{S}}{}^{(k)}| \cdot |\mathcal{R}_{\mathcal{S}}{}^{(k)}|)/(|\mathcal{T}| \times |\mathcal{R}|) = \eta_k$, $\forall k \in \{1, .., K\}$

[2-] Sound beam-pairs from the training set for each user, $\mathcal{K} := \mathcal{T}_{\mathcal{S}}{}^{(k)} \times \mathcal{R}_{\mathcal{S}}{}^{(k)}$. The sounding leads to obtaining the SINR values, as formulated in (7.3).

[3-] Generate SINR data-set tensor $\mathbf{S}$, in (7.5).

[4-] Select BA approach: generalized matrix completion or tensor completion:

if Generalized point-to-point model selected **then**

- Split $\mathbf{S}$ tensor into K incomplete matrices.
- Fine-tune MLP and AE based on the available training samples in $\mathcal{K}$, respecting the overhead ratio η_k for each user k, to obtain the optimal weights of the neural networks.
- Use optimal weights to predict for samples in $\mathcal{L}$ in order to complete the SINR matrix for each user.

else

- Fine-tune CNN based on the available training samples in $\mathcal{K}$, corresponding to the total overhead η for all users, to obtain the optimal weights of the neural network.
- Use optimal weights to predict for samples in $\mathcal{L}$ in order to complete the SINR tensor for all users.

end if

[5-] Select the optimal codebook indexes based on the highest SINR, in (7.4).

Output: $\mathbf{F}_{u^\star}$, $\mathbf{W}_{i^\star}$

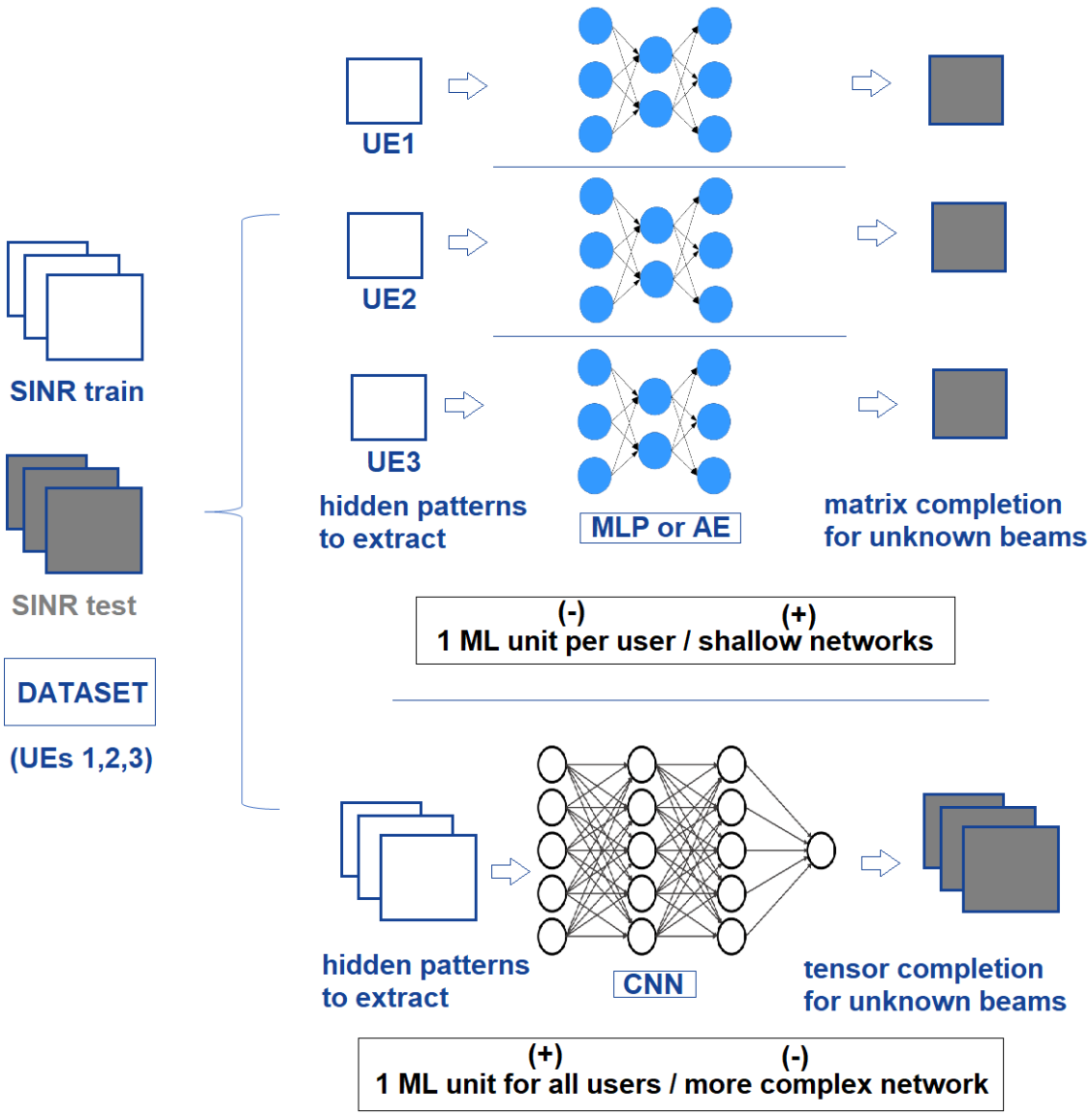


Figure 7.2: Proposed solution for multi user system with K=3: generalized point-to-point vs tensor completion

System configuration for all proposed models	
System-parameter	Numerical value
number of antennas N_T at UEs	$64, 128, 256, 512, 1024$
number of antennas N_R at BS	$64, 128, 256, 512, 1024$
codebook cardinality $\|\mathcal{T}\|$ at UEs	$64, 128, 256, 512, 1024$
codebook cardinality $\|\mathcal{R}\|$ at BS	$64, 128, 256, 512, 1024$
number of users	$4, 6, 8, 16$
overhead ratio η regime	$0.5, 0.3, 0.1$
number of $OFMD$ sub-carriers N_c	64
number of channel paths: $rank(\mathbf{G})$	$1, 2, 3, 4$ (randomly attributed to users)
transmitted power P_u (W)	1
MLP, CNN, AE number of layers	$1, 2, 3, 4$
MLP, CNN, AE number of neurons per layer	$32, 64, 128, 256, 512, 1024$
MLP, CNN, AE batch size	$8, 32, 128, 256$
MLP, CNN, AE learning rate	$10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$
AE bottleneck dimension	$2, 8, 16$
CNN kernel-filter dimension	2

Table 7.1: Proposed system parameters and hyperparameters

7.6 Numerical simulations:

In our comprehensive experimental protocol, we systematically explored a wide range of system parameters to gain deep insights into the performance of our proposed multi-user beam alignment algorithm. The study involved varying the number of antennas at both user equipments and base stations across multiple configurations: 64, 128, 256, 512, and 1024. Additionally, we investigated different codebook cardinalities at UEs and BS, ranging from 64 to 1024, and considered varying numbers of users, from 4 to 16. To understand the impact of channel characteristics, we examined different numbers of channel paths, randomly attributed to users, ranging from 1 to 4, to modelize different experimental situation for multiple users (LoS, NLoS..). The experimental setup also included variations in the overhead ratio regime, exploring values of 0.5, 0.3, and 0.1. We standardized the number of OFDM sub-carriers at 64 , the carrier frequency at 60 GHz and the considered transmitted power is fixed at 1W. Thus, the SINR values are in range [0, 10 dB]. Our investigation delved into the architecture of neural networks, varying the number of layers and neurons per layer for MLP, CNN, and AE models. Additionally, we explored different batch sizes and learning rates for these networks. The optimal hyperparameters of all proposed neural networks are obtained from the offline grid-search cross-validation. Consequently, this meticulous exploration allowed us to comprehensively analyze the algorithm's behavior under diverse operational scenarios, providing valuable insights into its adaptability and efficiency across a spectrum of real-world conditions. The experimental protocol is resumed in table (7.1).

Regarding the QoS evaluation metrics, we use the training MSE to evaluate the

training error for all models, expressed as:

$$\text{Train MSE}_{u,i}^{(k)} = \frac{1}{|\mathcal{K}|} \sum_{(u,i)\in\mathcal{K}} (\widehat{SINR}_{u,i}^{(k)} - SINR_{u,i}^{(k)})^2 \quad , \forall k \in \{1, .., K\} \tag{7.8}$$

where $SINR_{u,i}^{(k)}$ is the true SINR value from (7.5) and $\widehat{SINR}_{u,i}^{(k)}$ is the predicted SINR value from our proposed neural networks, for all samples in $\mathcal{K}$. For test samples in $\mathcal{L}$, we similarly define:

$$\text{Test MSE}_{u,i}^{(k)} = \frac{1}{|\mathcal{L}|} \sum_{(u,i)\in\mathcal{L}} (\widehat{SINR}_{u,i}^{(k)} - SINR_{u,i}^{(k)})^2 \quad , \forall k \in \{1, .., K\} \tag{7.9}$$

Finally, we introduce γ, as the total number of required training samples, defined as the product of the number of antennas at UEs and BS, the number of users K and the total signaling overhead ratio η:

$$\gamma = |\mathcal{T}| \times |\mathcal{R}| \times K \times \eta \tag{7.10}$$

One of the research directions of this chapter is to investigate the impact of increasing γ on the quality of prediction of proposed ML tools for both approaches.

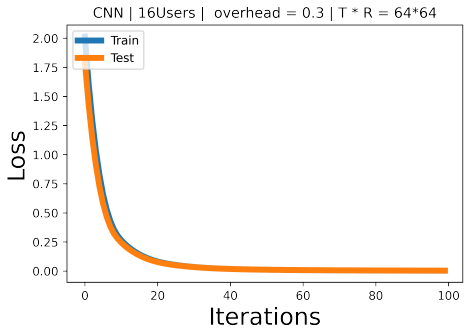
7.6.1 Primary results

The CNN demonstrated remarkable efficiency in the task of tensor completion across various configurations. The training and test Mean Squared Error values consistently trended toward zero throughout the iterations, underscoring the efficacy of the model. For instance, in Figure (7.3a), the CNN effectively extracted features from 16 users and a base station, each equipped with 64 antennas, requiring only 30% of the total training samples to achieve precise completion. A similar trend was observed in Figure (7.3b), where approximately 30% of the total training samples sufficed to achieve MSE values around 10^{-3}. As the MIMO dimensions increased, such as with 4 users and a BS, each with 256 antennas, the optimal signaling overhead decreased to 10%, resulting in MSE values of 10^{-4} as the total number of training sample increased.

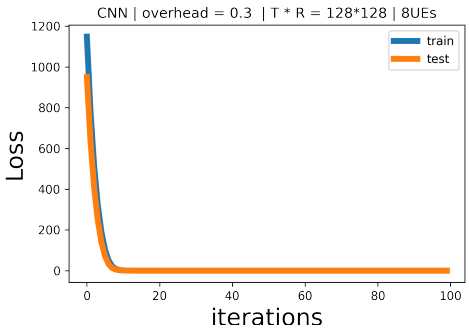
However, certain simulations revealed subtle overfitting tendencies in the CNN learning curves which impacts model's robustness (particularly evident when we tested transitioning to lower Signal-to-Noise Ratio regimes, outside the proposed experimental protocol). Addressing these challenges, the next section outlines research directions aimed at refining and enhancing the model's robustness.

Thus, the tensor completion approach based on CNN not only preserves 90% of available training samples with high accuracy but also maintains a moderate level of computational complexity, making it a promising solution for multi-user real-world applications.

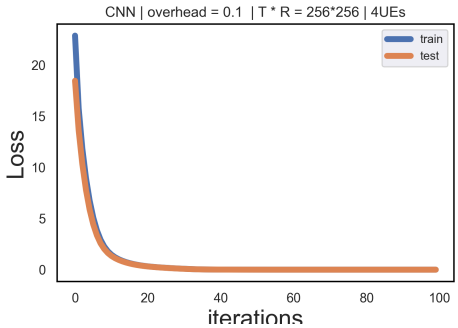
Similarly to CNN, the AE succeeded with the partial BA mission with 30% and 10% of total samples with balanced learning curves with no symptoms of overfitting/underfitting. The range of MSE values reaches 10^{-2} overall and jumps to



(a) 64×64 CNN Learning curve for $K = 16, \eta = 0.3$

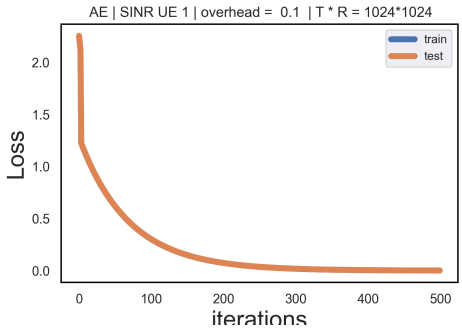


(b) 128×128 CNN Learning curve for $K = 8, \eta = 0.3$

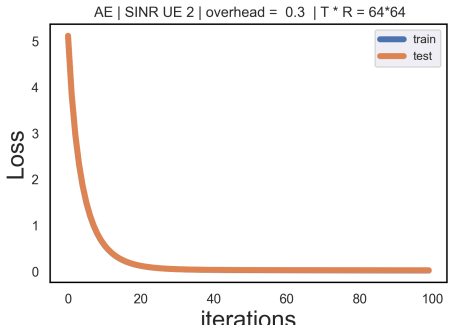


(c) 256×256 CNN Learning curve for $K = 4, \eta = 0.1$

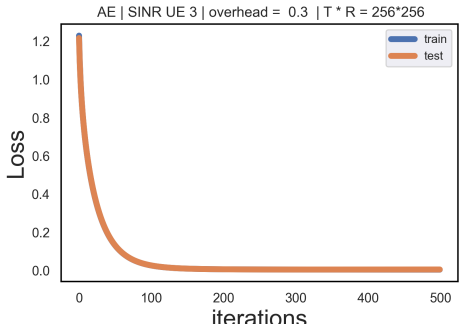
Figure 7.3: CNN Learning curves for multi user Beam Alignment



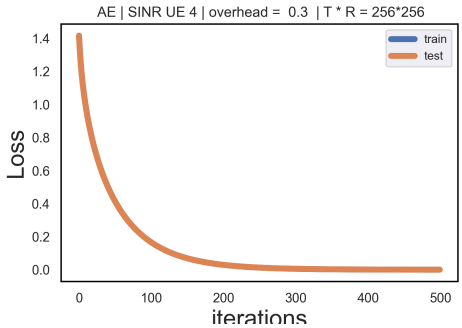
(a) 1024×1024 AE Learning curve for $K = 6, \eta = 0.1$ for User 1



(b) 64×64 AE Learning curve for $K = 16, \eta = 0.3$ for User 2



(c) 256×256 AE Learning curve for $K = 16, \eta = 0.3$ for User 3

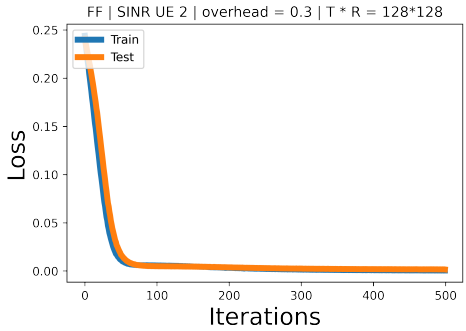


(d) 256×256 AE Learning curve for $K = 16, \eta = 0.3$ for User 4

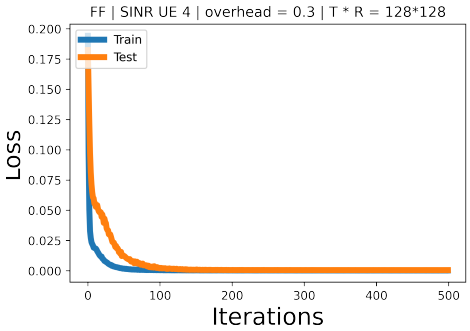
Figure 7.4: AE Learning curves for multi user Beam Alignment

around 10^{-3} for extremely high dimensional MIMO setups. In (7.4a), we focus on the first UE among 6 simulated: with 10% overhead ratio, this highest MIMO configuration, guarantee the best performance from QoS perspective with the lowest MSE values for training and test samples. In (7.4b), the second UE among 16 had just 64 antennas, which represent the opposite case where we have the lowest γ. Indeed, the optimal overhead ratio reaches 30% and the MSE is ranging around 10^{-1} for training and test samples. In (7.4c) and (7.4d), when we compare the performance between different users, we empirically observe similar behaviour and results in terms of learning curves, required optimal overhead, required optimal combination of hyperparameters and reached error values. However, some exceptions are notices where AE failed to accurately complete the SINR matrix, even with higher number of training samples: these cases state one limitation for AE in processing some users, characterized by "difficult" experimental situations (NLoS with 4 channel paths, too far from BS or simply exposed to much noise and interference). Finally, it is primordial to notice that AE is the fastest and less complex neural architecture proposed in the whole PhD and in this chapter particularly. Thus, AE have the smallest dimensions and number of parameters, making it less greedy to computational resources. With accurate predictions and low signaling overhead ratios, AE excels with its accuracy/complexity compromise.

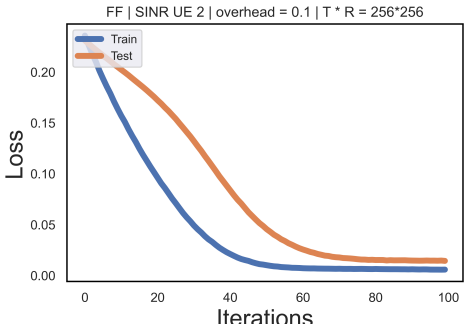
Conversely, numerical results suggest that MLP is the most balanced and robust



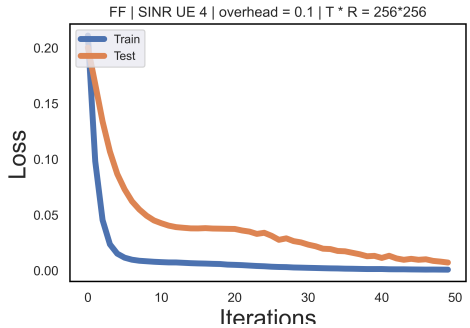
(a) 128×128 MLP Learning curve for $K = 8, \eta = 0.3$ for User 2



(b) 128×128 MLP Learning curve for $K = 8, \eta = 0.3$ for User 4



(c) 256×256 MLP Learning curve for $K = 4, \eta = 0.1$ for User 2



(d) 256×256 MLP Learning curve for $K = 4, \eta = 0.1$ for User 4

Figure 7.5: MLP Learning curves for multi user Beam Alignment

model, with higher complexity than AE and lower complexity than CNN. The Feed Forward architecture guarantee considerably low MSE values for training and test samples, ranging around 10^{-4} and scores the best performances among all when it handles the largest proposed system configurations. In (7.5a) and (7.5b), users illustrate similar performances, loss values, monotonic convergence of the corresponding learning curves, ideal combinations of hyperparameters and the optimal required overhead ratio. As long as we increase the system configurations and so, γ, MLP in (7.5c) and (7.5d), training and test MSE decrease to around 10^{-3}.

To sum up, the proposed models succeeded in the Partial BA for multi user with optimal overhead ratios equal to 30% for small configurations and reaches 10% for the largest proposed MIMO setups. Figure (7.6) tracks the -log(MSE) in function of total number of training samples. For the sake of simplicity, we consider the average between the very close training and test MSE values to represent the MSE loss and γ is calculated as in (7.10):

- CNN is the largest proposed model, according to the mission in question, where completing a sparse tensor with low number of samples is a harder task compared to matrix completion for each user separately, due to the increased dimensionality and the intricate patterns that can exist in higher-order, especially when the dataset includes much noise and interference between users. Consequently, CNN guarantee better performance as long as we increase γ,

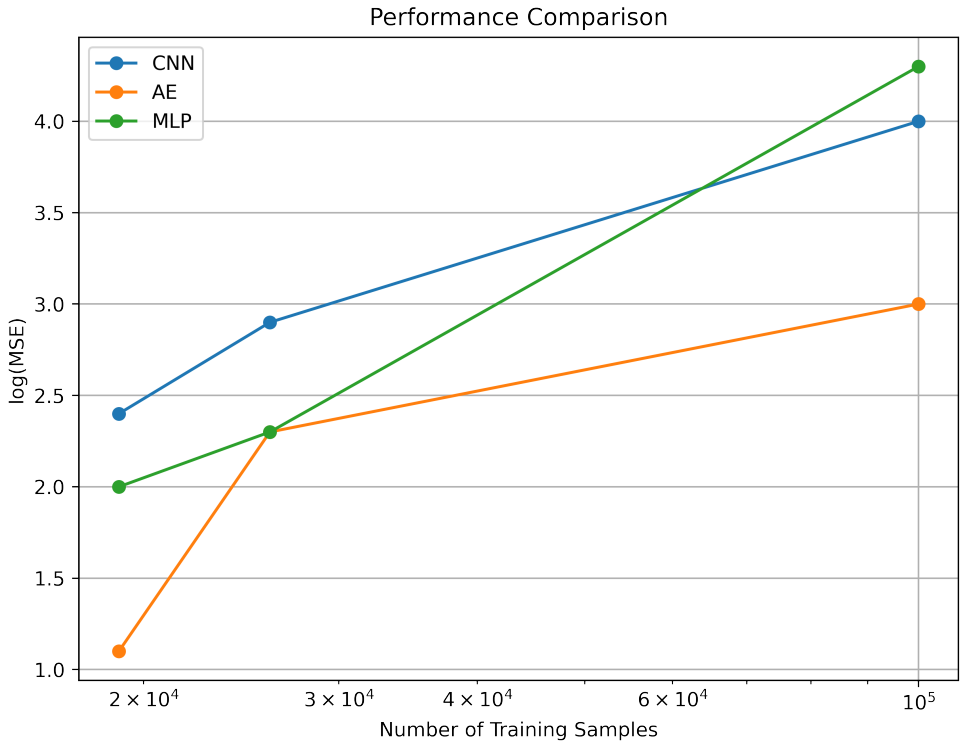


Figure 7.6: QoS models evaluation: -log(MSE) in function of the number of training samples

as shown in figure (7.6): -log(MSE) jumps from 2 for $\gamma = 2 \times 10^4$ to 4 for γ values beyond 10^5.

- AE is the smallest proposed model, the easiest to fine-tune, the fastest in terms of cross-validation and predictions. However, it conducts the worst performance relatively with the highest training and test MSE values among all. Besides, more training samples for AE means better predictions, where -log(MSE) jumps from 1.1 for $\gamma = 2 \times 10^4$ to 3 for $\gamma = 10^5$. The inflection point marks the intersection between AE and MLP performances at around $\gamma = 3.7 \times 10^4$. Beyond this γ value, the slope of the curve drastically increases for MLP and decreases for AE.

- Feed Forward architecture also depends on more data to decrease training and test error. Thus, it is the best model when the number of antennas at both sides of the transmission become extremely high. At $\gamma = 6.5 \times 10^4$, MLP reaches very close loss values for training and test samples compared to the CNN. Beyond this value, MLP overpasses all models and illustrate the best QoS performance reaching 4.3 for 10^5 training samples.

7.6.2 Limitations and perspectives

Regarding the primary results, the slightly presence of overfitting in the CNN performance and the failure of AE and MLP to complete matrices of rare users in "difficult" experimental situations, lead to following research directions:

- Given the neural networks' high sensitivity to even minor shifts in optimal hyperparameters, it becomes imperative to expand our exploration. Increasing the granularity of our grid search and experimenting with larger combinations of hyperparameters is crucial. While this approach might incur a higher cross-validation latency, the potential pay-off is substantial: discovering more robust hyperparameter combinations that yield significantly lower test Mean Squared Error. This expanded search necessitates a deeper refinement process, involving careful considerations such as the selection of appropriate activation functions and the incorporation of dropout techniques. These architectural enhancements hold the promise of overcoming the limitations observed in our initial results. By meticulously refining these elements, we aim to fortify the neural networks, making them more resilient to variations and ensuring the reliability and accuracy of our Beam Alignment models.

- From a machine learning perspective, the adage "more data, better predictions" holds particularly true in the realm of multi-user Beam Alignment. As the number of users increases, so does the volume of data at our disposal. This abundance of data opens doors to enhanced prediction quality, a fundamental principle in machine learning. Leveraging this principle, our proposed approaches can be adapted to embrace a transfer learning paradigm. In this paradigm, our models undergo pre-training across a myriad of experimental

use-cases and datasets. Subsequently, these pre-trained models are fine-tuned to tackle new real-time matrix/tensor completion challenges. This approach capitalizes on the wealth of historical data, allowing our models to learn and adapt swiftly to evolving scenarios.

- Furthermore, the proliferation of users equipped with large dimensional MIMO configurations introduces an intriguing prospect: Distributed Learning. In this paradigm, the training process and feature extractions are distributed among multiple models. These models collaboratively learn from past performances, with knowledge rigorously shared and distributed across different ML units. While this methodology converges towards a concept akin to Federated Learning, it also prompts critical questions about the orchestration between massive users and their corresponding voluminous datasets. Properly managed and optimized, this distributed approach has the potential to outperform our initial proposed methods, transcending their limitations and leading to more efficient and effective Beam Alignment strategies.

7.7 Conclusion

In this pivotal chapter, we embarked on the exploration of partial Beam Alignment in a multi-user scenario, a terrain brimming with challenges and possibilities. We laid out the groundwork, first presenting the intricacies of the fully-analog system architecture and formulating the problem. Within this complex landscape, we delineated two approaches to tackle the issue at hand. Our dataset, comprising SINR values for each beam-pair across all users, served as the foundation upon which our Alignment procedure was constructed, employing DFT sub-sampled codebooks. The chapter unfolded with the unveiling of our primary numerical results, painting a promising picture: our proposed machine learning tools not only fulfilled their mission but did so with a low pilot overhead ratio, showcasing the efficiency of both proposed approaches. As we delved deeper, the chapter culminated in a critical QoS-focused comparison of the models' performances in function of the available training samples, stated some limitations and expressed research directions to combat them. As we prepare to embark on the conclusive chapter, these findings illuminate the path forward, guiding us towards a comprehensive understanding of the project's context, the contributions made, the results obtained, and the perspectives that lie ahead.

Chapter 8

Conclusions and perspectives

"We cannot solve our problems with the same thinking we used when we created them."

Albert Einstein.

8.1 Conclusions

In the pursuit of enhancing spectral efficiency for 5G networks, mmWave MIMO technology emerged, offering significant advancements through advanced precoding techniques. Despite its potential, the complexity of real-world urban environments and the unique characteristics of mmWave frequencies pose challenges. Massive MIMO communication at mmWave frequencies requires precise beam alignment, vital for establishing robust initial links. Classical methods in conventional standards, like WiGig, involve exhaustive beam sounding, our benchmark, leading to excessive pilot-signaling overhead and the impossibility to deploy it in large dimensional MIMO applications. Our research addresses this issue by proposing Partial and Blind Beam Alignment, an approach integrating machine learning techniques. By leveraging sub-sampled codebooks and employing neural networks and matrix factorization, we aim to reduce pilot overhead and accurately identify optimal beam pairs.

In the literature, SotA methods are divided into two families, classical BA and ML based BA. The first approaches rely on the brute-force BA and are generally based on hierarchical codebooks, Compressed Sensing, Beam Coding and multiple other tools, aiming to optimize the BA process using all available samples. They generally require CSI-based channel estimation and hybrid beamforming architectures. On the other hand, ML relies on less training samples with promising results. However, the investigation of its complexity and hardware requirements illustrate the challenges behind applying AI tools in Wireless Communication systems.

In this context, this work aims to investigate the feasibility of the proposed ML-based Beam Alignment approach, relying on low-complexity fully-analog architectures with limited RF chains in and shallow neural networks. Thus, we started from

basic Uplink point-to-point configuration and build up, step-by-step, with continuous formulation of the new encountered technical problems and constraints, aiming to mathematically and empirically answer these problematics. The main findings of this PhD work are:

- We first considered a point-to-point, narrowband, LoS, Uplink scenario using one RF chain for UE and one RF chain for BS. The solution for the BA problem is based on the application of Matrix Factorization and its variants to fulfill the task of Blind and Partial BA using sub-sampled codebooks. In addition to the theoretical guarantees, the numerical results serve as compelling evidence, demonstrating the efficacy of our approach that seamlessly integrates model-based and data-driven methodologies. Most notably, our method accomplishes its objectives with efficiency, utilizing a mere 10% of the available beams and achieving a fully CSI-blind solution. This accomplishment represents a significant stride in addressing the challenge of large signaling overhead in beam alignment and has led to the publication of the WCNC conference paper in [86].

- Our second finding in the PhD journey commenced with an extension of our system model, elevating the complexity of our experimental setup to mirror real-world conditions, including NLoS wideband model, multiple RF chains at BS and sub-sampled DFT codebooks. Following this, we introduced the Multi-Layer Perceptron architecture, presenting its input-output equations, problem statement, and the formulated solution. Thus, we proposed a comparative study, illuminating the nuanced interplay between complexity and prediction quality for both methods: our findings underscore that a mere 10% of the total beam pairs prove sufficient for accurately aligning the beams between User Equipment and Base Station for both methodologies in a point-to-point single user scenario. Furthermore, we discerningly explored the similarities and differences in models behavior under varying transmitted power scenarios. These outcomes illuminate the practical viability of our methods, providing a solid foundation for their application in real-world contexts and ushering in a new era of more efficient and effective Beam Alignment in future communication systems. These findings had allowed the submission of the journal paper in [87].

- The third outcome of this PhD delves deep into the intricate quantization in reference to the practical constraints of ML models deployment. We commenced with the establishment of the system architecture and the formulation of model equations. A mathematical formulation of these constraints was undertaken, paving the way for the creation of the corresponding quantized Received Signal Energy dataset. The learning approach was then elucidated, mapping out the cascade of Binary Logistic Regression layers and delineating their respective input-output equations at each stage. Notably, the numerical revelations highlighted a consistent and assertive finding: our optimal overhead ratio (maintained at 10% throughout the entirety of this thesis) can seamlessly

harmonize with an efficient one-bit binary quantization scheme. This adjustment, remarkably, doesn't compromise the predictive quality while upholding stringent low-complexity prerequisites, thereby affirming the practicality of our proposed approach, illustrated in the publication of the ICC conference paper in [88].

- Finally, we laid the foundation of the proposed fully-analog system architecture for multi-user Beam Management, formulating the underlying problem and system equations. Within this complex framework, we outlined two approaches to address the issue. Our dataset, a comprehensive collection of SINR values for each beam-pair across all users, became the cornerstone upon which our Alignment procedure was crafted, leveraging DFT sub-sampled codebooks. As our exploration deepened, we uncovered primary numerical results: shallow CNN, MLP and AE not only met their objectives but did so with a low pilot overhead ratio, underscoring the efficiency of both proposed approaches. In addition, the experimental protocol involved a quality-of-service-focused comparison, exploring the models' performances in function of the available training samples. In acknowledging the limitations encountered, we charted paths for future research, highlighting directions to enhance our methodologies and overcome these challenges.

8.2 Perspectives

The application of AI tools in Wireless Communication systems is in the early stages of exploration. As an increasing number of studies highlight its significance, especially in addressing challenges related to Beam Alignment and Management, fresh opportunities are emerging. These developments are paving the way for inventive methodologies that have the potential to bring about unforeseen shifts in the existing paradigms.

Regarding our specific challenge, several potential project avenues come into focus. Thus, we distinguish two families of research and development directions:

Wireless Communication perspectives:

- Exploring the impact of user mobility and velocity represents a natural progression for our research, particularly in real-life scenarios. Addressing this challenge could involve delving into recurrent neural networks, like the Long Short Term Memory architecture, or transitioning towards a Reinforcement Learning paradigm, allowing us to adapt to dynamic user movements.

- Evaluating our models on industrial datasets such as DeepMIMO [58], Quadriga [59] or DeepSense [60] can provide valuable real-world validation.

- Exploring matrix/tensor completion for partial Beam Alignment using Reflective Intelligent Surfaces, an emerging trend in mmWave MIMO literature, offers both challenges and promising results that warrant investigation.

- Optimal design of codebooks for Partial Beam Alignment tasks is crucial. Transitioning from conventional equispaced beams, typically used in DFT codebooks, to more optimized schemes could substantially enhance our approach. Similarly, optimizing power allocation to users, perhaps through techniques like water-filling, represents another avenue for refinement.

Machine Learning perspectives:

- Enhancing the robustness of our system equations by incorporating additional factors is a promising avenue. This can involve striking a balance between complexity and accuracy using advanced AI tools and methodologies. Techniques like Transfer Learning, leveraging pre-trained models on extensive datasets, or adopting Federated Learning with distributed training and prediction across multiple coordinated ML units, could provide significant insights.

- Expanding the scope of our offline grid-search cross-validation could significantly enhance the quality of predictions made by our ML tools.

- Compressing the dimensions of these AI tools, akin to the principles of tinyML, quantizing their weights, and reducing the number of parameters, is vital for future industrial deployment and efficiency.

- Enhancing the interpretability of ML models is an interesting research direction. Opening the black boxes behind ML tools, understanding the underlying features, and explaining the predictions made by neural networks are rigorous tasks that can yield valuable insights across various AI applications in Wireless Communications and beyond.

Appendix A

Proof: BCD convergence

We prove that the two necessary conditions for the convergence of BCD are satisfied:

- the loss function is strongly convex, per block, i.e., we should show that sub-problem S1 and S2 have a unique solution
- the constraints of the MF prob $\boldsymbol{\theta}_u \in \mathbb{R}^d$, $\boldsymbol{\psi}_i \in \mathbb{R}^d$, are separable and individually convex

Recall that the sub-problem S1 is expressed as:

$$(S1) : \boldsymbol{\theta}_u^{(k+1)} =_{\boldsymbol{\theta}_u \in \mathbb{R}^d} \quad [-2\boldsymbol{\theta}_u^T \mathbf{r}_u^{(k)} + \boldsymbol{\theta}_u^T (\ \mathbf{Q}_u^{(k)} + \mu_u \mathbf{I}_D\ ) \boldsymbol{\theta}_u] \ = \ f_1(\boldsymbol{\theta}_u), \quad \forall u, \quad (e.1) \tag{A.1}$$

Afterwards, we prove that the equivalent form in (e.1), is a strongly convex function: $f_1(\boldsymbol{\theta}_u)$ is strongly in $\boldsymbol{\theta}_u$. To that end, we derive the corresponding Hessian:

$$\nabla^2 f_1(\boldsymbol{\theta}_u) := 2(\ \mathbf{Q}_u^{(k)} + \mu_u \mathbf{I}_D\ ), \quad \forall u, \quad (g.1)$$

For the Hessian expression in (g.1), $\mathbf{Q}_u^{(k)} \succeq \mathbf{0}$ is by definition a Positive Semi Definite matrix, $\mu_u \mathbf{I} \succ \mathbf{0}$ is a Positive Definite matrix, and $(\ \mathbf{Q}_u^{(k)} + \mu_u \mathbf{I}_D\ ) \succ \mathbf{0}$ is PD matrix. Thus, the Hessian is 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 sub-problem (S1) is unique. The sub-problem (S2) is written as:

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

Subsequently, we show that the equivalent form in (e.2), is a strongly convex function: $f_2(\boldsymbol{\psi}_i)$ is strongly in $\boldsymbol{\psi}_i$. Therefore, we calculate the corresponding Hessian:

$$\nabla^2 f_2(\boldsymbol{\psi}_i) := 2(\ \mathbf{P}_i^{(k+1)} + {\lambda_i}^{(i)} \mathbf{I}_D\ ), \quad \forall i, \quad (g.2) \tag{A.3}$$

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

Appendix B

Proof: BLR convergence

We prove the following statements:

- 1.A) $f(\mathbf{w})$ in (6.10) is a strongly convex function in w.

- 1.B) The Gradient Descent iterations in (B.2) are monotonically decreasing, i.e., $f(\mathbf{w}_{t+1}) \leq f(\mathbf{w}_t), \forall t \in \mathbb{N}$, and converge to the global optimum $\mathbf{w}^*$, i.e., $\lim_{t\to\infty}\{\mathbf{w}_t\}_t = \mathbf{w}^*$.

Where:
(6.10):

$$\mathbf{w}^* := f(\mathbf{w}) = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \mathcal{L}(\mathbf{w}) + \lambda\|\mathbf{w}\|_2^2 =$$

$$\frac{1}{|\mathcal{K}|ln(2)} \sum_{(u,i)\in\mathcal{T}_\mathcal{S}\times\mathcal{R}_\mathcal{S}} log_2(1 + exp^{-(\mathbf{w}^T\mathbf{x}_{u,i}Q_{u,i}))}) + \lambda\|\mathbf{w}\|_2^2 \tag{B.1}$$

(B.2):

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha_t\nabla_\mathbf{w} f(\mathbf{w}_t), \forall t \in \{1, .., N_{epochs}\} \tag{B.2}$$

Proof 1.A):

i) show that $\mathcal{L}_{u,i}(\mathbf{w})$ in (6.8) is convex. Given $\mathcal{L}_{u,i}(\mathbf{w}) = c.log_2(1 + exp(-\mathbf{w}^T\mathbf{a}))$ where $\mathbf{a} = \mathbf{x}_{u,i}.Q_{u,i}$, and $c = (ln(2))^{-1}, c > 0$. Moreover, $exp(-\mathbf{w}^T\mathbf{a}) = g(\mathbf{w})$ is convex in $\mathbf{w}$ and $log_2(1 + x) = h(x)$ is monotonically increasing in x. Thus, $\mathcal{L}_{u,i}(\mathbf{w}) = c.h(g(\mathbf{w}))$ is convex, since it is a composition of a monotonically increasing $h(-)$ and convex functions $g(-)$, and $c > 0$ constant. The sum of $\mathcal{L}_{u,i}(\mathbf{w})$ convex functions is convex which proves the convexity of $\mathcal{L}(\mathbf{w})$ in (6.9).

ii) $f(\mathbf{w})$ is the sum of a convex function $\mathcal{L}(\mathbf{w})$, and a strongly convex function (regularization term $\lambda\|\mathbf{w}\|_2^2$). Therefore, $f(\mathbf{w})$ in (6.10) is strongly convex function in $\mathbf{w}$.

Proof 1.B): the proof follows from $f(\mathbf{w})$ being strongly convex.

Bibliography

[1] Wang, Yi and Wei, Zhiqing and Feng, Zhiyong, Beam Training and Tracking in MmWave Communication: A Survey, arXiv, 10.48550/ARXIV.2205.10169, 2022.

[2] IEEE Std 802.15.3c-2009. IEEE Standard, Oct 2009.

[3] IEEE Std 802.11ad-2012. IEEE Standard, Dec 2012.

[4] 3GPP, "TS 38.211 V16.7.1 NR; Physical channels and modulation."

[5] Noh, Song and Zoltowski, Michael D. and Love, David J. Multi-Resolution Codebook and Adaptive Beamforming Sequence Design for Millimeter Wave Beam Alignment, pp. 5689-5701, 2017.

[6] Kokshoorn, Matthew and Chen, He (Henry) and Wang, Peng and Li, Yonghui and Vucetic, Branka, Millimeter Wave MIMO Channel Estimation Using Overlapped Beam Patterns and Rate Adaptation, 2016.

[7] Y. M. Tsang, A. S. Y. Poon and S. Addepalli, "Coding the Beams: Improving Beamforming Training in mmWave Communication System," 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011, Houston, TX, USA, 2011, pp. 1-6, doi: 10.1109/GLOCOM.2011.6134486, 2011.

[8] S. Buzzi and C. D'Andrea, "Subspace Tracking and Least Squares Approaches to Channel Estimation in Millimeter Wave Multiuser MIMO," in IEEE Transactions on Communications, vol. 67, no. 10, pp. 6766-6780, Oct. 2019, doi: 10.1109/TCOMM.2019.2924885, 2019.

[9] E. Khordad, I. B. Collings, S. V. Hanly and G. Caire, "Compressive Sensing Based Beam Alignment Schemes for Time-Varying Millimeter-Wave Channels," in IEEE Transactions on Wireless Communications, doi: 10.1109/TWC.2022.3205702, 2022.

[10] Hadi Ghauc, Mikael Skoglund, Hossein Shokri-Ghadikolaei, Carlo Fischione, Ali H. Sayed, 'Learning Kolmogorov Models for Binary Random Variables, 2018.

[11] Yetis, Cenk M. and Björnson, Emil and Giselsson, Pontus, Joint Analog Beam Selection and Digital Beamforming in Millimeter Wave Cell-Free Massive MIMO Systems, arXiv, 10.48550/ARXIV.2103.11199, 2021.

[12] Dreifuerst, Ryan M. and Heath, Robert W. and Yazdan, Ali, Massive MIMO Beam Management in Sub-6 GHz 5G NR, arXiv, 10.48550/ARXIV.2204.06064, April 2022.

[13] Ma, Ke and He, Dongxuan and Sun, Hancun and Wang, Zhaocheng and Chen, Sheng, Deep Learning Assisted Calibrated Beam Training for Millimeter-Wave Communication Systems, arXiv, 10.48550/ARXIV.2101.05206, 2021.

[14] Nguyen, Khuong N. and Ali, Anum and Mo, Jianhua and Ng, Boon Loong and Va, Vutha and Zhang, Jianzhong Charlie, Beam Management with Orientation and RSRP using Deep Learning for Beyond 5G Systems, arXiv, 10.48550/ARXIV.2202.02247, Feb 2022.

[15] Aldalbahi, A.; Shahabi, F.; Jasim, M. BRNN-LSTM for Initial Access in Millimeter Wave Communications. Electronics 2021, 10, 1505. https://doi.org/10.3390/electronics10131505, 2021.

[16] Dehkordi, Saeid K. and Kobayashi, Mari and Caire, Giuseppe, Adaptive Beam Tracking based on Recurrent Neural Networks for mmWave Channels, arXiv, https://doi.org/10.48550/arxiv.2108.04548, 2021.

[17] Hussain, Muddassar and Michelusi, Nicolo, Learning and Adaptation for Millimeter-Wave Beam Tracking and Training: a Dual Timescale Variational Framework, arXiv, https://doi.org/10.48550/arxiv.2107.05466, 2021.

[18] Ryan M. Dreifuerst and Samuel Daulton and Yuchen Qian and Paul Varkey and Maximilian Balandat and Sanjay Kasturia and Anoop Tomar and Ali Yazdan and Vish Ponnampalam and Robert W. Heath, Optimizing Coverage and Capacity in Cellular Networks using Machine Learning, arXiv, 2010.13710, arXiv:2010.13710, Feb 2021.

[19] N. Narengerile, J. Thompson, P. Patras and T. Ratnarajah, "Deep Reinforcement Learning-Based Beam Training for Spatially Consistent Millimeter Wave Channels," 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Helsinki, Finland, pp. 579-584, doi: 10.1109/PIMRC50174.2021.9569732, 2021.

[20] L. Wang, B. Ai, Y. Niu, M. Gao and Z. Zhong, "Adaptive Beam Alignment Based on Deep Reinforcement Learning for High Speed Railways," 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, pp. 1-6, doi: 10.1109/VTC2022-Spring54318.2022.9860897, 2022.

[21] R. Barathy and P. Chitra, "Applying Matrix Factorization In Collaborative Filtering Recommender Systems," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 635-639, doi: 10.1109/ICACCS48705.2020.9074227, 2020.

[22] A. A. M. Saleh and R. Valenzuela, "A Statistical Model for Indoor Multipath Propagation," in IEEE Journal on Selected Areas in Communications, vol. 5, no. 2, pp. 128-137, doi: 10.1109/JSAC.1987.1146527, February 1987.

[23] Cai, Ling Xu, Jun Liu, Ju Pei, Tao. Integrating spatial and temporal contexts into a factorization model for POI recommendation. International Journal of Geographical Information Science. 32. 1-23. 10.1080/13658816.2017.1400550, 2017.

[24] "Euclidean Algorithm", https://en.wikipedia.org/wiki/Euclideanalgorithm. Accessed: 2023-09-22.

[25] "Timeline of Algorithms", https://en.wikipedia.org/wiki/Timelineofalgorithms. Accessed: 2023-09-22.

[26] "Charles Babbage", https://en.wikipedia.org/wiki/CharlesBabbage. Accessed: 2023-09-22.

[27] "Ada Lovelace.", https://en.wikipedia.org/wiki/AdaLovelace. Accessed: 2023-09-22.

[28] "Alan Turing." https://en.wikipedia.org/wiki/AlanTuring. Accessed: 2022-08-31.

[29] The university of Queensland Australia, "The history of Artificial Intelligence", https://qbi.uq.edu.au/brain/intelligent-machines/history-artificial-intelligence, Accessed: 2023-09-22.

[30] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848, 2009.

[31] Zenan Ling and Haotian Ma and Yu Yang and Robert C. Qiu and Song-Chun Zhu and Quanshi Zhang, Explaining AlphaGo: Interpreting Contextual Effects in Neural Networks, arXiv, 1901.02184, 2019.

[32] Mingyu Zong and Bhaskar Krishnamachari, a survey on GPT-3, arXiv, 2212.00857, 2022.

[33] Dennett, D.C . Can Machines Think?. In Teuscher, C. (eds) Alan Turing: Life and Legacy of a Great Thinker. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-05642-4, 2004.

[34] Fjodor Van Veen, "Neural Networks zoo prequel: cells and layers", published on March 31, 2017, https://www.asimovinstitute.org/author/fjodorvanveen/, Accessed: 2023-09-22.

[35] Project Pro Authors, "Introduction to Convolutional Neural Networks Architecture", published on July 15, 2023, https://www.projectpro.io/article/introduction-to-convolutional-neural-networks-algorithm-architecture/560, Accessed: 2023-09-22.

[36] Mansini, R., Ogryczak, W., Speranza, M.G. (2015). Linear Models for Portfolio Optimization. In: Linear and Mixed Integer Programming for Portfolio Optimization. EURO Advanced Tutorials on Operational Research. Springer, Cham. https://doi.org/10.1007/978-3-319-18482, 2015.

[37] Ming-Hua Lin, Jung-Fa Tsai, Chian-Son Yu, "A Review of Deterministic Optimization Methods in Engineering and Management", DOI: 10.1155/2012/756023, June 2012.

[38] Dylan McGrath, "Overcome 5G mmWave measurement issues", published on June 28, 2021, https://www.5gtechnologyworld.com/overcome-5g-mmwave-measurement-issues/, Accessed: 2023-09-22.

[39] Ibrahim Hemadeh, Satyanarayana Katla, Mohammed El-Hajjar, L. Hanzo, "Millimeter-Wave Communications: Physical Channel Models, Design Considerations, Antenna Constructions and Link-Budget", DOI: 10.5258/SOTON/D0344, December 2017.

[40] Emil Bjornson, "multiple antenna communications", published in github emilbjornson on August 31, 2021, https://www.github.com/emilbjornson/multipleantennacommunications, Accessed: 2023-09-22..

[41] C. B. A. Wael, Suyoto, N. Armi, A. S. Satyawan, B. E. Sukoco and A. Subekti, "Performance of Regularized Zero Forcing (RZF) Precoding for Multiuser Massive MIMO-GFDM System over mmWave Channel," 2021 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), Bandung, Indonesia, 2021, pp. 256-259, doi: 10.1109/ICRAMET53537.2021.9650347, 2021.

[42] T. Kebede, Y. Wondie, J. Steinbrunn, H. B. Kassa and K. T. Kornegay, "Precoding and Beamforming Techniques in mmWave-Massive MIMO: Performance Assessment," in IEEE Access, vol. 10, pp. 16365-16387, 2022, doi: 10.1109/ACCESS.2022.3149301, 2022.

[43] Chunhua Zhu, Qinwen Ji, Xinying Guo, Jiankang Zhang, "Mmwave massive MIMO: one joint beam selection combining cuckoo search and ant colony optimization", EURASIP Journal on Wireless Communications and Networking, https://doi.org/10.1186/s13638-023-02272, July 2023.

[44] O. Alluhaibi, M. Nair, A. Hazzaa, A. Mihbarey and J. Wang, "3D Beamforming for 5G Millimeter Wave Systems Using Singular Value Decomposition and Particle Swarm Optimization Approaches," 2018 International Conference on

Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2018, pp. 15-19, doi: 10.1109/ICTC.2018.8539578, 2018.

[45] H. Huang, Y. Song, J. Yang, G. Gui and F. Adachi, "Deep-Learning-Based Millimeter-Wave Massive MIMO for Hybrid Precoding," in IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 3027-3032, March 2019, doi: 10.1109/TVT.2019.2893928, 2019.

[46] Khormuji, Majid Nasiri and Renaud-Alexandre Pitaval. "Statistical beam codebook design for mmWave massive MIMO systems." 2017 European Conference on Networks and Communications (EuCNC): 1-5, 2017.

[47] R. W. Heath, N. González-Prelcic, S. Rangan, W. Roh and A. M. Sayeed, "An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems," in IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 3, pp. 436-453, April 2016, doi: 10.1109/JSTSP.2016.2523924, 2016.

[48] A. Alkhateeb, O. El Ayach, G. Leus and R. W. Heath, "Channel Estimation and Hybrid Precoding for Millimeter Wave Cellular Systems," in IEEE Journal of Selected Topics in Signal Processing, vol. 8, no. 5, pp. 831-846, Oct. 2014, doi: 10.1109/JSTSP.2014.2334278, 2014.

[49] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi and R. W. Heath, "Spatially Sparse Precoding in Millimeter Wave MIMO Systems," in IEEE Transactions on Wireless Communications, vol. 13, no. 3, pp. 1499-1513, March 2014, doi: 10.1109/TWC.2014.011714.130846, 2014.

[50] A. Meijerink and A. F. Molisch, "On the Physical Interpretation of the Saleh–Valenzuela Model and the Definition of Its Power Delay Profiles," in IEEE Transactions on Antennas and Propagation, vol. 62, no. 9, pp. 4780-4793, Sept. 2014, doi: 10.1109/TAP.2014.2335812, 2014.

[51] A. Khalili, S. Rangan and E. Erkip, "On Single-User Interactive Beam Alignment in Next Generation Systems: A Deep Learning Viewpoint," 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 2021, pp. 1-6, doi: 10.1109/ICCWorkshops50388.2021.9473733, 2021.

[52] B. W. Domae, R. Li and D. Cabric, "Machine Learning Assisted Phase-less Millimeter-Wave Beam Alignment in Multipath Channels," 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 2021, pp. 1-7, doi: 10.1109/GLOBECOM46510.2021.9685678, 2021.

[53] A. N. Uwaechia and N. M. Mahyuddin, "A Comprehensive Survey on Millimeter Wave Communications for Fifth-Generation Wireless Networks: Feasibility and Challenges," in IEEE Access, vol. 8, pp. 62367-62414, 2020, doi: 10.1109/ACCESS.2020.2984204, 2020.

[54] Lyutianyang Zhang and Sumit Roy, "Optimal Beam Training for mmWave Massive MIMO using 802.11ay", arXiv 2211.15990, 2022.

[55] Matthieu Roy, Stephane Paquelet, Matthieu Crussière. Degrees of Freedom of Ray-Based Models for mm-Wave Wideband MIMO-OFDM. 2019 IEEE Global Communications Conference (GLOBECOM), Dec 2019, Waikoloa, United States, 2019.

[56] Y. Heng and J. G. Andrews, "Machine Learning-Assisted Beam Alignment for mmWave Systems," in IEEE Transactions on Cognitive Communications and Networking, vol. 7, no. 4, pp. 1142-1155, Dec. 2021, doi: 10.1109/TCCN.2021.3078147, 2021.

[57] Ahmed Alkhateeb, "DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Application", arXiv 1902.06435, 2019.

[58] S. Jaeckel, L. Raschkowski, K. Börner and L. Thiele, "QuaDRiGa: A 3-D Multi-Cell Channel Model With Time Evolution for Enabling Virtual Field Trials," in IEEE Transactions on Antennas and Propagation, vol. 62, no. 6, pp. 3242-3256, June 2014, doi: 10.1109/TAP.2014.2310220, 2014.

[59] A. Alkhateeb et al., "DeepSense 6G: A Large-Scale Real-World Multi-Modal Sensing and Communication Dataset," in IEEE Communications Magazine, vol. 61, no. 9, pp. 122-128, September 2023, doi: 10.1109/MCOM.006.2200730, 2023.

[60] A. Taha, M. Alrabeiah and A. Alkhateeb, "Enabling Large Intelligent Surfaces With Compressive Sensing and Deep Learning," in IEEE Access, vol. 9, pp. 44304-44321, 2021, doi: 10.1109/ACCESS.2021.3064073, 2021.

[61] A. Alkhateeb, J. Mo, N. Gonzalez-Prelcic and R. W. Heath, "MIMO Precoding and Combining Solutions for Millimeter-Wave Systems," in IEEE Communications Magazine, vol. 52, no. 12, pp. 122-131, December 2014, doi: 10.1109/MCOM.2014.6979963, 2014.

[62] R. Zhang, H. Zhang, W. Xu and X. You, "A Closed-Form PS-DFT Codebook Design for mmWave Beam Alignment," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, pp. 1-6, doi: 10.1109/ICC.2019.8761632, 2019.

[63] Junyeol Hong and Hyeonjin Chung and Sunwoo Kim, "Experimental Demonstration of Location-aware Beam Alignment", arXiv 2003.03053, 2020.

[64] Junyi Wang, Z Lan, C Pyo, "Beam Codebook Based Beamforming Protocol for Multi-Gbps mmWave WPAN systems", IEEE Journal on Selected Areas in Communications, vol 27, pp. 1390-1399, Oct. 2009.

[65] Sooyoung Hur and Taejoon Kim and David J. Love and James V. Krogmeier and Timothy A. Thomas and Amitava Ghosh, "Millimeter Wave Beamforming for Wireless Backhaul and Access in Small Cell Networks", Transactions on Communications, vol. 61, pp.4391-4403, Oct. 2013.

[66] C. Qi, K Chen, O. Dobre, "Hierarchical Codebook-Based Multi user Beam Training for mmWave Massive MIMO", IEEE Transactions on Wireless Communications, vol. 19, no. 12; Dec. 2020.

[67] K Chen and C. Qi, "Beam Training Based on Dynamic Hierarchical codebook for mmWave Massive MIMO", IEEE Communications Letters, vol. 23, no. 1, Jan. 2019.

[68] T. Oh, C. Song, J. Jung, "A new RF Beam Training Method and Asymptotic Performance Analysis for Multi User mmWave Systems", IEEE Access, vol. 6, pp. 48125-48135, 2018.

[69] T. Kim and D.J. Love, "Virtual AoA and AoD Estimation for Sparse mmWave MIMO channels", proc. IEEE Workshop on Signal Processing advances in Wireless Communications (SPAWC), pp. 146-150, 2015.

[70] L. Yan, X. Fang, L. Hao, "A Fast Beam Alignment Scheme for Dual-Band HSR Wireless Networks", IEEE Transactions on Vehicular Technology, vol. 69, no. 4, Apr. 2020.

[71] V. Va, T. Shimizun G. Bansal, "Online Learning for Position-Aided mmWave Beam Training", IEEE Access, vol. 7, 2019.

[72] W. Zhang, W. Zhang, J. Wu, "UAV Beam Alignment for Highly Mobile mmWave Communications", IEEE Transactions of Vehicular technology, vol. 96, Aug. 2020.

[73] K. Satyanarayana, M. El-Hajjar, A. Mourad, "Deep Learning Aided Fingerprint based Beam Alignment for mmWave Vehicular Communication", IEEE Transactions on Vehicular Technology, vol. 68, no. 11, Nov. 2019.

[74] A. Alkhateeb, S. Alex, P. Varkey, "Deep Learning Coordinated Beamforming for Highly-mobile mmWave Systems", IEEE Access, vol. 6, 2018.

[75] J. Zhang, Y. Huang, Y. Zhou, "Beam Alignment and Tracking for mmWave Communications via Bandit Learning", IEEE Transactions of Communications, vol. 68, no.9, Sep. 2020.

[76] Aldalbahi, A.; Shahabi, F.; Jasim, M. BRNN-LSTM for Initial Access in Millimeter Wave Communications. Electronics 2021, 10, 1505. https://doi.org/10.3390/electronics10131505, 2021.

[77] K. N. Nguyen, A. Ali, J. Mo, B. L. Ng, V. Va and J. C. Zhang, "Beam Management with Orientation and RSRP using Deep Learning for Beyond 5G Systems," 2022 IEEE International Conference on Communications Workshops (ICC Workshops), Seoul, Korea, Republic of, 2022, pp. 133-138, doi: 10.1109/ICCWorkshops53468.2022.9814507, 2022.

[78] C. M. Yetis, E. Björnson and P. Giselsson, "Joint Analog Beam Selection and Digital Beamforming in Millimeter Wave Cell-Free Massive MIMO Systems," in IEEE Open Journal of the Communications Society, vol. 2, pp. 1647-1662, 2021, doi: 10.1109/OJCOMS.2021.3094823, 2021.

[79] R. M. Dreifuerst, R. W. Heath and A. Yazdan, "Massive MIMO Beam Management in Sub-6 GHz 5G NR," 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 2022, pp. 1-5, doi: 10.1109/VTC2022-Spring54318.2022.9860458, 2022.

[80] R. M. Dreifuerst et al., "Optimizing Coverage and Capacity in Cellular Networks using Machine Learning," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 2021, pp. 8138-8142, doi: 10.1109/ICASSP39728.2021.9414155, 2021.

[81] L. Qianrui, "Hybrid Precoding for Wideband Multi-user MIMO Millimeter Wave System," 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 2019, pp. 1-6, doi: 10.1109/WCNC.2019.8885454, 2019.

[82] X. Zhou, H. Liu, B. Wang, J. Huang and Y. Wang, "Tensor Dictionary Manifold Learning for Channel Estimation and Interference Elimination of Multi-User Millimeter-Wave Massive MIMO Systems," in IEEE Access, vol. 10, pp. 5343-5358, 2022, doi: 10.1109/ACCESS.2021.3128929, 2022.

[83] Florian Muhr and Lorenzo Zaniboni and Saeid K. Dehkordi and Fernando Pedraza Nieto and Giuseppe Caire, "Beam Alignment with an Intelligent Reflecting Surface for Integrated Sensing and Communication", arXiv 2304.01848, 2023.

[84] Jide Yuan and George C. Alexandropoulos and Eleftherios Kofidis and Tobias Lindstrm Jensen and Elisabeth De Carvalho, "Tensor-based Channel Tracking for RIS-Empowered Multi-User MIMO Wireless Systems", arXiv 2202.08315, 2022.

[85] A. Ktari, H. Ghauch and G. Rekaya, "Matrix Factorization for Blind Beam Alignment in Massive mmWave MIMO," 2022 IEEE Wireless Communications and Networking Conference (WCNC), 2022, pp. 2637-2642, doi: 10.1109/WCNC51071.2022.9771772, Austin, Texas, 2022.

[86] A. Ktari, H. Ghauch and G. Rekaya, "Machine Learning techniques for blind Beam Alignment in mmWave Massive MIMO", Eurasip Journal on Wireless Communications and Networking, 2023.

[87] A. Ktari, H. Ghauch and G. Rekaya, "Cascaded Binary Classifiers for Blind Beam Alignment in mmWave MIMO Using One-Bit Quantization," 2023 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 80-85, doi: 10.1109/ICCWorkshops57953.2023.10283648, Rome, Italy, 2023.

[88] W. Xu, F. Gao, X. Tao, J. Zhang and A. Alkhateeb, "Computer Vision Aided mmWave Beam Alignment in V2X Communications," in IEEE Transactions on Wireless Communications, vol. 22, no. 4, pp. 2699-2714, doi: 10.1109/TWC.2022.3213541, April 2023.

[89] J. Chen et al., "Hybrid Beamforming/Combining for Millimeter Wave MIMO: A Machine Learning Approach," in IEEE Transactions on Vehicular Technology, vol. 69, no. 10, pp. 11353-11368, Oct. 2020, doi: 10.1109/TVT.2020.3009746, 2020.

[90] K. Satyanarayana, M. El-Hajjar, A. A. M. Mourad and L. Hanzo, "Deep Learning Aided Fingerprint-Based Beam Alignment for mmWave Vehicular Communication," in IEEE Transactions on Vehicular Technology, vol. 68, no. 11, pp. 10858-10871, Nov. 2019, doi: 10.1109/TVT.2019.2939400, 2019.

[91] A. L. Makara, B. T. Csath'o, L. Csurgai-Horv'ath and B. P. Horv'ath, "Measurement-based Indoor Beam Alignment Utilizing Deep Learning," 2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 2021, pp. 1-6, doi: 10.1109/ICECCME52200.2021.9590951, 2021.

[92] J. Zhang, Y. Huang, J. Wang, X. You and C. Masouros, "Intelligent Interactive Beam Training for Millimeter Wave Communications," in IEEE Transactions on Wireless Communications, vol. 20, no. 3, pp. 2034-2048, March 2021, doi: 10.1109/TWC.2020.3038787, 2021.

[93] Umut Demirhan and Ahmed Alkhateeb, "Integrated Sensing and Communication for 6G: Ten Key Machine Learning Roles", arXiv, eess.SP, 2208.02157, 2022.

[94] G. Callebaut, F. Rottenberg, L. V. der Perre and E. G. Larsson, "Grant-Free Random Access of IoT devices in Massive MIMO with Partial CSI," 2023 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6, doi: 10.1109/WCNC55385.2023.10118929, Glasgow, United Kingdom, 2023.

[95] Y. Heng and J. G. Andrews, "Machine Learning-Assisted Beam Alignment for mmWave Systems," 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1-6, 2019.

[96] C. Qi, Y. Wang and G. Y. Li, "Deep Learning for Beam Training in Millimeter Wave Massive MIMO Systems," in IEEE Transactions on Wireless Communications, 2020.

[97] Yuyang Wang, Nitin Jonathan Myers, Nuria González-Prelcic and Robert W. Heath Jr, Deep Learning-based Compressive Beam Alignment in mmWave Vehicular Systems, 2021.

[98] Guillaume Larue, "Modeles IA pour le traitement des signaux numeriques dans le contexte des futurs reseaux 6G-IoT", Télécom Paris, IP Paris, 15 October 2022.

[99] I. J. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, http://www.deeplearningbook.org, 2016.

[100] Han Yan, Benjamin W. Domae, and Danijela Cabric. 2020. MmRAPID: Machine Learning assisted Noncoherent Compressive Millimeter-Wave Beam Alignment. In Proceedings of the 4th ACM Workshop on Millimeter-Wave Networks and Sensing Systems (mmNets'20), 2020.

[101] K. Ma, D. He, H. Sun, Z. Wang and S. Chen, "Deep Learning Assisted Calibrated Beam Training for Millimeter-Wave Communication Systems," in IEEE Transactions on Communications, vol. 69, no. 10, pp. 6706-6721, Oct. 2021, doi: 10.1109/TCOMM.2021.3098683, 2021.

[102] J. Zhang and C. Masouros, "Learning-Based Predictive Transmitter-Receiver Beam Alignment in Millimeter Wave Fixed Wireless Access Links," in IEEE Transactions on Signal Processing, vol. 69, pp. 3268-3282, 2021, doi: 10.1109/TSP.2021.3076899, 2021.

[103] S. Rezaie, J. Morais, E. de Carvalho, A. Alkhateeb and C. N. Manchón, "Location- and Orientation-aware Millimeter Wave Beam Selection for Multi -Panel Antenna Devices," GLOBECOM, pp. 597-602, doi:10.1109/GLOBECOM48099.2022.10001089, Rio de Janeiro, Brazil, 2022.

[104] W. Ma, C. Qi and G. Y. Li, "Machine Learning for Beam Alignment in Millimeter Wave Massive MIMO," in IEEE Wireless Communications Letters, vol. 9, no. 6, pp. 875-878, June 2020.

[105] Y. Heng and J. G. Andrews, "Machine Learning-Assisted Beam Alignment for mmWave Systems," 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1-6.

[106] C. Qi, Y. Wang and G. Y. Li, "Deep Learning for Beam Training in Millimeter Wave Massive MIMO Systems," in IEEE Transactions on Wireless Communications.

[107] Han Yan, Benjamin W. Domae, and Danijela Cabric. 2020. MmRAPID: Machine Learning assisted Noncoherent Compressive Millimeter-Wave Beam Alignment. In Proceedings of the 4th ACM Workshop on Millimeter-Wave Networks and Sensing Systems (mmNets'20) Oct. 2021.

[108] J. Zhang and C. Masouros, "Learning-Based Predictive Transmitter-Receiver Beam Alignment in Millimeter Wave Fixed Wireless Access Links," in IEEE Transactions on Signal Processing, vol. 69, pp. 3268-3282, 2021, doi: 10.1109/TSP.2021.3076899.

[109] W. Ma, C. Qi and G. Y. Li, "Machine Learning for Beam Alignment in Millimeter Wave Massive MIMO," in IEEE Wireless Communications Letters, vol. 9, no. 6, pp. 875-878, June 2020, doi: 10.1109/LWC.2020.2973972.

[110] Sohrabi, Foad and Chen, Zhilin and Yu, Wei, "Deep Active Learning Approach to Adaptive Beamforming for mmWave Initial Alignment"

[111] J. Song, J. Choi and D. J. Love, "Common Codebook Millimeter Wave Beam Design: Designing Beams for Both Sounding and Communication With Uniform Planar Arrays," in IEEE Transactions on Communications, vol. 65, no. 4, pp. 1859-1872, April 2017, doi: 10.1109/TCOMM.2017.2665497.

[112] Wu, Wen and Cheng, Nan and Zhang, Ning and Yang, Peng and Zhuang, Weihua and Xuemin, Fast mmwave Beam Alignment via Correlated Bandit Learning.

[113] Duan, Qiyou and Kim, Taejoon and Ghauch Hadi, Enhanced Beam Alignment for Millimeter Wave MIMO Systems: A Kolmogorov Model, 2020.

[114] Nitin Jonathan Myers and Amine Mezghani and Robert W. Heath, "Swift-Link: A Compressive Beam Alignment Algorithm for Practical mmWave Radios", 2019.

[115] F. Maschietti, D. Gesbert, P. de Kerret and H. Wymeersch, "Robust Location-Aided Beam Alignment in Millimeter Wave Massive MIMO," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, 2017, pp. 1-6, doi: 10.1109/GLOCOM.2017.8254901.

[116] A. Abdallah, A. Celik, M. M. Mansour and A. M. Eltawil, "Deep Learning Based Frequency-Selective Channel Estimation for Hybrid mmWave MIMO Systems," in IEEE Transactions on Wireless Communications, doi: 10.1109/TWC.2021.3124202.

[117] M. N. Khormuji and R. Pitaval, "Statistical beam codebook design for mmWave massive MIMO systems," 2017 European Conference on Networks and Communications (EuCNC), 2017, pp. 1-5, doi: 10.1109/EuCNC.2017.7980681.

[118] Raj, Vishnu and Nayak, Nancy and Kalyani, Sheetal, "Deep Reinforcement Learning based Blind mmWave MIMO Beam Alignment", 2020.

[119] Myers, Nitin Jonathan and Wang, Yuyang and González-Prelcic, Nuria and Heath, Robert W. "Deep learning-based beam alignment in mmWave vehicular networks", 2019.

[120] J. Chen et al., "Hybrid Beamforming/Combining for Millimeter Wave MIMO: A Machine Learning Approach," in IEEE Transactions on Vehicular Technology, vol. 69, no. 10, pp. 11353-11368, Oct. 2020, doi: 10.1109/TVT.2020.3009746.

[121] K. Satyanarayana, M. El-Hajjar, A. A. M. Mourad and L. Hanzo, "Deep Learning Aided Fingerprint-Based Beam Alignment for mmWave Vehicular Communication," in IEEE Transactions on Vehicular Technology, vol. 68, no. 11, pp. 10858-10871, Nov. 2019, doi: 10.1109/TVT.2019.2939400.

[122] A. L. Makara, B. T. Csath'o, L. Csurgai-Horv'ath and B. P. Horv'ath, "Measurement-based Indoor Beam Alignment Utilizing Deep Learning," 2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 2021, pp. 1-6, doi: 10.1109/ICECCME52200.2021.9590951.

[123] J. Zhang, Y. Huang, J. Wang, X. You and C. Masouros, "Intelligent Interactive Beam Training for Millimeter Wave Communications," in IEEE Transactions on Wireless Communications, vol. 20, no. 3, pp. 2034-2048, March 2021, doi: 10.1109/TWC.2020.3038787.

[124] M. Wang, F. Gao, S. Jin and H. Lin, "An Overview of Enhanced Massive MIMO With Array Signal Processing Techniques," in IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 5, pp. 886-901, Sept. 2019, doi: 10.1109/JSTSP.2019.2934931.

[125] W. Huang, Y. Huang, Y. Zeng and L. Yang, "Wideband Millimeter Wave Communication With Lens Antenna Array: Joint Beamforming and Antenna Selection With Group Sparse Optimization," in IEEE Transactions on Wireless Communications, vol. 17, no. 10, pp. 6575-6589, Oct. 2018, doi: 10.1109/TWC.2018.2860963.

[126] Z. Gong, F. Jiang and C. Li, "Angle Domain Channel Tracking With Large Antenna Array for High Mobility V2I Millimeter Wave Communications," in IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 5, pp. 1077-1089, Sept. 2019, doi: 10.1109/JSTSP.2019.2933352.

[127] J. Wang, R. Han, L. Bai, T. Zhang, J. Liu and J. Choi, "Coordinated Beamforming for UAV-Aided Millimeter-Wave Communications Using GPML-Based Channel Estimation," in IEEE Transactions on Cognitive Communications and Networking, vol. 7, no. 1, pp. 100-109, March 2021, doi: 10.1109/TCCN.2020.3048399.

[128] J. Zhang, Y. Huang, J. Wang and X. You, "Intelligent Beam Training for Millimeter-Wave Communications via Deep Reinforcement Learning," 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1-7, doi: 10.1109/GLOBECOM38437.2019.9014113.

[129] L. Zhu, J. Zhang, Z. Xiao and X. Cao, "User Fairness Non-orthogonal Multiple Access (NOMA) in Millimeter-Wave Communications," 2018 IEEE/CIC International Conference on Communications in China (ICCC Workshops), 2018, pp. 80-84, doi: 10.1109/ICCChinaW.2018.8674488.

[130] B. Yang, Z. Yu, J. Lan, R. Zhang, J. Zhou and W. Hong, "Digital Beamforming-Based Massive MIMO Transceiver for 5G Millimeter-Wave Communications," in IEEE Transactions on Microwave Theory and Techniques, vol. 66, no. 7, pp. 3403-3418, July 2018, doi: 10.1109/TMTT.2018.2829702.

[131] Z. Xiao, L. Zhu, Z. Gao, D. O. Wu and X. -G. Xia, "User Fairness Non-Orthogonal Multiple Access (NOMA) for Millimeter-Wave Communications With Analog Beamforming," in IEEE Transactions on Wireless Communications, vol. 18, no. 7, pp. 3411-3423, July 2019, doi: 10.1109/TWC.2019.2913844.

[132] R. G. Stephen and R. Zhang, "Uplink Channel Estimation and Data Transmission in Millimeter-Wave CRAN With Lens Antenna Arrays," in IEEE Transactions on Communications, vol. 66, no. 12, pp. 6542-6555, Dec. 2018, doi: 10.1109/TCOMM.2018.2859996.

[133] Q. Lu, T. Lin and Y. Zhu, "Channel Estimation and Hybrid Precoding for Millimeter Wave Communications: A Deep Learning-Based Approach," in IEEE Access, vol. 9, pp. 120924-120939, 2021, doi: 10.1109/ACCESS.2021.3108625.

[134] P. Wang, J. Fang, X. Yuan, Z. Chen and H. Li, "Intelligent Reflecting Surface-Assisted Millimeter Wave Communications: Joint Active and Passive Precoding Design," in IEEE Transactions on Vehicular Technology, vol. 69, no. 12, pp. 14960-14973, Dec. 2020, doi: 10.1109/TVT.2020.3031657.

[135] A. Sethi and R. V. Raja Kumar, "Channel Estimation using Approximate Conjugate Gradient Pursuit for Hybrid MIMO System in Millimeter Wave Communication," 2020 IEEE India Council International Subsections Conference (INDISCON), 2020, pp. 236-241, doi: 10.1109/INDISCON50162.2020.00056

[136] Y. Luo, Y. Yang, G. Zhen, D. He and L. Zhang, "Machine Learning based Analog Beam Selection for Concurrent Transmissions in mmWave Heterogeneous Networks," 2021 IEEE/CIC International Conference on Communications in China (ICCC), 2021, pp. 788-792, doi: 10.1109/ICCC52777.2021.9580272.

[137] Marco Giordani, Michele Polese, Arnab Roy, Douglas Castor, Michele Zorzi, "A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies", Published in 5G NR on March 19, 2018, https://medium.com/5g-nr/mobility-with-mm-waves-4b2085b83d91, Accessed: 2023-09-22.

[138] H. Kim, S. Moon and I. Hwang, "Machine Learning-based Channel Tracking for Next-Generation 5G Communication System," 2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN), 2021, pp. 267-269, doi: 10.1109/ICUFN49451.2021.9528722.

[139] X. Wang and M. Cenk Gursoy, "Multi-Agent Double Deep Q-Learning for Beamforming in mmWave MIMO Networks," 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, 2020, pp. 1-6, doi: 10.1109/PIMRC48278.2020.9217114.

[140] L. Li et al., "Millimeter-Wave Networking in the Sky: A Machine Learning and Mean Field Game Approach for Joint Beamforming and Beam-Steering," in IEEE Transactions on Wireless Communications, vol. 19, no. 10, pp. 6393-6408, Oct. 2020, doi: 10.1109/TWC.2020.3003284.

[141] A. Alkhateeb, I. Beltagy and S. Alex, "MACHINE LEARNING FOR RELIABLE MMWAVE SYSTEMS: BLOCKAGE PREDICTION AND PROACTIVE HANDOFF," 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2018, pp. 1055-1059, doi: 10.1109/GlobalSIP.2018.8646438.

[142] Y. Zhang, M. Alrabeiah and A. Alkhateeb, "Learning Beam Codebooks with Neural Networks: Towards Environment-Aware mmWave MIMO," 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2020, pp. 1-5, doi: 10.1109/SPAWC48557.2020.9154320.

[143] D. Kim and N. Lee, "Machine Learning based Detections for mmWave Two-Hop MIMO Systems using One-Bit Transceivers," 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2019, pp. 1-5, doi: 10.1109/SPAWC.2019.8815577

[144] A. Pan, T. Zhang and X. Han, "Location information aided beam allocation algorithm in mmWave massive MIMO systems," 2017 IEEE/CIC International Conference on Communications in China (ICCC), 2017, pp. 1-6, doi: 10.1109/ICCChina.2017.8330410.

[145] M. Alrabeiah and A. Alkhateeb, "Deep Learning for mmWave Beam and Blockage Prediction Using Sub-6 GHz Channels," in IEEE Transactions on Communications, vol. 68, no. 9, pp. 5504-5518, Sept. 2020, doi: 10.1109/TCOMM.2020.3003670.

ECOLE DOCTORALE

Titre : Apprentissage automatique pour l'Alignement des faisceaux pour les systèmes MIMO massifs à ondes millimétriques

Mots clés : Alignment des faisceaux, Massive MIMO, Apprentissage automatique, Apprentissage profond, Ondes millimétriques

Résumé : La demande croissante en efficacité spectrale, stimulée par les exigences strictes des réseaux 5G, a accéléré le développement de la technologie MIMO en ondes millimétriques, offrant des améliorations architecturales significatives grâce à des techniques de précodage avancées. Cette technologie présente des gains substantiels en termes d'efficacité spectrale et énergétique par rapport aux systèmes MIMO traditionnels. Cependant, le potentiel transformateur du MIMO en mmWave est entravé par les réalités complexes des environnements urbains réels et les propriétés physiques complexes inhérentes aux fréquences des ondes millimétriques.
De manière cruciale, dans les communications massives MIMO en mmWave, le beamforming et le combining jouent des rôles essentiels : la large bande passante et la fréquence de fonctionnement élevée des systèmes à ondes millimétriques nécessitent un beamforming/combining dans le domaine analogique, rendant les approches entièrement digitales techniquement impossibles. Au cœur du MIMO massif en mmWave se trouve le problème d'Alignement des Faisceaux, exigeant l'identification des paires de faisceaux d'émission et de réception optimales qui maximisent le rapport signal/bruit, assurant ainsi une liaison initiale robuste.
Les normes existantes, telles que WiGig, utilisent des méthodes exhaustives de sondage des faisceaux, testant chaque paire de faisceaux possible pour trouver celle qui maximise le SNR. Cependant, cela entraîne un surcoût important de signalisation de pilotes: le principal problème que nous cherchons à résoudre tout au long de cette thèse de doctorat. Notre recherche révolutionne l'Alignement des Faisceaux en intégrant des techniques de pointe en apprentissage automatique pour l'Alignement Partiel des Faisceaux, réduisant considérablement les surcharge de pilotes en ne sondant qu'un sous-ensemble de paires de faisceaux à l'aide de codebooks sous-échantillonnés. Ainsi, nous exploitons les énergies des signaux reçus à partir de ces sondages de paires de faisceaux, en utilisant des réseaux neuronaux peu profonds, la factorisation matricielle et leurs variantes pour résoudre avec précision des problèmes de régression non-linéaire et logistique, cruciaux pour déterminer la qualité des paires de faisceaux restantes. Un objectif fondamental de cette thèse est de déterminer la complexité de l'échantillonnage pour ces méthodes d'apprentissage automatique. Cette complexité dicte le nombre minimum d'échantillons d'entraînement nécessaires pour un apprentissage efficace et une transmission fiable. Nous examinons également les performances des modèles ML proposés sans estimation préalable du canal, introduisant le concept d'Alignement Aveugle des Faisceaux, ouvrant ainsi la voie à un changement radical de paradigme. De plus, notre recherche explore en profondeur les subtilités de la quantification, une contrainte pratique vitale. Nous explorons ensuite des compromis cruciaux : identifier la surcharge minimale correspondant au schéma de quantification optimal tout en investiguant le compromis classique entre précision et complexité. Grâce à une progression méthodologique systématique, allant des scénarios point-à-point basiques à bande étroite aux complexes architectures multi-utilisateurs à large bande, cette thèse de doctorat offre des insights et des solutions précieuses: les contributions proposées font progresser les domaines des communications en mmWave et les applications d'apprentissage automatique dans les systèmes sans fil, surpassant les benchmarks existants et affrontant les limites des approches conventionnelles.

Title : Machine Learning for beam Alignment in mmWave massive MIMO

Keywords : Beam Alignment, Massive MIMO, Machine Learning, Deep Learning, mmWave

Abstract : The escalating demand for spectral efficiency driven by the stringent requirements of 5G networks has spurred the development of mmWave MIMO technology, promising significant architectural improvements through advanced precoding techniques. This technology presents substantial gains in spectral and energy efficiencies compared to traditional MIMO systems. However, the transformative potential of mmWave MIMO is hampered by the complex realities of real-world urban environments and the intricate physical properties inherent to mmWave frequencies.

Crucially, in mmWave massive MIMO communication, beamforming and combining play pivotal roles: the high bandwidth and operating frequency of mmWave systems necessitate analog domain beamforming/combining, rendering fully digital approaches technically non feasible. At the heart of mmWave large-dimensional MIMO lies the Beam Alignment problem, requiring the identification of optimal transmit and receiver beam pairs that maximize the Signal-to-Noise ratio, ensuring a robust initial link.

Existing standards, such as WiGig, employ exhaustive beam sounding methods, testing each possible beam pair to find the one maximizing SNR. Consequently, it leads to substantial pilot-signaling overhead, the major problem we aim to encounter throughout this PhD. Our research revolutionizes Beam Alignment by integrating cutting-edge machine learning techniques for Partial Beam Alignment, significantly reducing the pilot overhead by soundings a subset of beam pairs using sub-sampled codebooks. Therefore, we leverage the received signal energies from these beam pairs soundings, employing shallow neural networks, matrix factorization, and their variants for accurately resolving non-linear and logistic regression problems, crucial for determining the quality of the remaining beam pairs.

A fundamental objective of this thesis is to determine the sample complexity for these machine learning methods. This complexity dictates the minimum number of training samples necessary for effective learning and reliable transmission. We delve into the performance of the proposed ML models without prior channel estimation, introducing the concept of Blind Beam Alignment, thus pioneering a paradigm shift. Furthermore, our research delves deep into the nuances of quantization, a vital practical constraint. We then explore critical compromises: identifying the minimum overhead ratio corresponding to the optimal quantization scheme on the one hand and navigating the classic trade-off between accuracy and complexity on the other hand.

Through systematic progression, ranging from basic point-to-point narrowband scenarios to intricate wideband multi-user architectures, this PhD thesis offers valuable insights and solutions. The proposed contributions advance the fields of mmWave communications and Machine Learning applications in wireless systems, outperforming existing benchmarks, and encountering the limitations of conventional approaches.

Institut Polytechnique de Paris
91120 Palaiseau, France