COLLEGE DOCTORAL BRETAGNE | MATHS, TELECOMS INFORMATIQUE, SIGNAL SYSTEMES, ELECTRONIQUE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

THÈSE DE DOCTORAT DE

L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Télécommunications*

Par
Baptiste CHATELIER

Model-based learning for high-dimensional wireless systems

Thèse présentée et soutenue à Rennes, le 30/01/2026
Unité de recherche : IETR (UMR CNRS 6164)
Thèse N° : 26ISAR 03 / D26 - 03

Rapporteurs avant soutenance :

Maxime Guillaud	Directeur de recherche, INRIA Lyon, France
Ghaya Rekaya-Ben Othman	Professeure des Universités, Télécom Paris, France

Composition du Jury :

Président :	Laurent Clavier	Professeur, IMT Nord Europe, France
Rapporteurs :	Maxime Guillaud	Directeur de recherche, INRIA Lyon, France
	Ghaya Rekaya-Ben Othman	Professeure des Universités, Télécom Paris, France
Examinateurs :	François Rottenberg	Assistant Professor, KU Leuven, Belgique
	Nir Shlezinger	Assistant Professor, Ben-Gurion University of the Negev, Israël
Dir. de thèse :	Matthieu Crussière	Professeur des Universités, INSA Rennes, France
Co-dir. de thèse :	Vincent Corlay	Chercheur industriel, Mitsubishi Electric R&D Centre Europe, France
Co-encadr. de thèse :	Luc Le Magoarou	Maître de Conférences, INSA Rennes, France

Invités :

Henk Wymeersch	Professor, Chalmers University of Technology, Suède

Doctoral Thesis

Model-based learning for high-dimensional wireless systems

Baptiste Chatelier
INSA Rennes and Mitsubishi Electric R&D Centre Europe, France

Jury composition

Laurent Clavier Professor, IMT Nord Europe, France	President
Maxime Guillaud Research Director, INRIA Lyon, France	Reviewer
Ghaya Rekaya-Ben Othman Professor, Télécom Paris, France	Reviewer
François Rottenberg Assistant Professor, KU Leuven, Belgium	External examiner
Nir Shlezinger Assistant Professor, Ben-Gurion University of the Negev, Israel	External examiner
Henk Wymeersch Professor, Chalmers University of Technology, Sweden	Invited examiner
Matthieu Crussière Professor, INSA Rennes, France	Thesis director
Vincent Corlay Industrial Researcher, Mitsubishi Electric R&D Centre Europe, France	Thesis co-director
Luc Le Magoarou Associate Professor, INSA Rennes, France	Thesis co-supervisor

This thesis is submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering, delivered by the MATISSE Doctoral School of the National Institute of Applied Sciences of Rennes.

January 30th, 2026

À ma grand-mère, Christiane

Note for the readers : In the pdf version of this manuscript, navigation is enabled through clickable links on chapters, sections, figures, tables, equations, and references.

Hope you will enjoy your reading !

Acknowledgements

Comme de nombreux doctorants avant moi, j'ai pu faire l'expérience que le doctorat, bien que finalisant un parcours de formation individuel, était avant tout une grande aventure collective. Aussi, je souhaite profiter de ces quelques mots pour remercier les (nombreuses) personnes ayant eu un lien avec la réalisation des travaux présentés dans ce manuscrit. Je m'excuse par avance auprès de tous ceux que j'aurais malencontreusement oubliés.

First of all, I would like to thank all the jury members for the interest they manifested for my work and for the very interesting exchanges we had during the defense. Je remercie Ghaya Rekaya Ben-Othman et Maxime Guillaud de m'avoir fait l'honneur d'être rapporteurs de ce (très long) manuscrit. Je remercie Laurent Clavier d'avoir accepté de présider le jury. I also would like to thank François Rottenberg and Nir Shlezinger for traveling to France to review my work, and for their really interesting questions. Finally, I would like to thank Henk Wymeersch for having given me the honor to serve as an invited jury member, even with his (very) busy schedule.

Je ne pourrais continuer ces remerciements sans mentionner l'exceptionnelle équipe d'encadrement de cette thèse. Matthieu, ces quelques mots ne sauront pas retranscrire à quel point tu as été important, aussi bien d'un point de vue professionnel que personnel. Tu m'as accueilli à bras ouverts dans le monde de la recherche il y a bientôt 5 ans, lorsque je suis venu toquer à ta porte pour trouver un stage. Pour cette raison, mais également pour ton encadrement exceptionnel, ainsi que ta pédagogie, je te suis à jamais reconnaissant. Vincent, en plus d'être un excellent encadrant, tu as su apporter une prise de recul parfois nécessaire à ces travaux. Pour cela, ainsi que ta présence pour faire le pont avec MERCE lorsque c'était nécessaire, je te remercie (promis, je ne mettrai plus de signe "environ égal" dans des théorèmes). Luc, tu as été au centre de l'encadrement de cette thèse. En plus de m'avoir proposé ce sujet particulièrement intéressant, tu as su orienter les travaux dans des directions toujours plus prometteuses, mais aussi recadrer lorsque je m'éparpillais trop. Pour ton excellence scientifique, ta présence au quotidien, mais également tes conseils avisés et ton sens de l'humour, je te remercie infiniment.

Je remercie MERCE d'avoir financé ces travaux, mais aussi d'avoir proposé un cadre si agréable pour leur bon déroulé. Je remercie tout particulièrement Nicolas Gresset, Sophie Pautonnier, Marie Plantard et Magali Branchereau pour leur précieuse aide administrative tout au long de la thèse.

I would like to thank Henk Wymeersch for having welcomed me in his group at Chalmers University of Technology during the 2024 winter. It was really an extraordinary

Swedish adventure, both professionally and personally, and I am looking forward to start the postdoc with you. I would also like to thank Christian Häger, Musa Furkan Keskin and José-Miguel Mateos Ramos from Chalmers for our mutual collaborations and hope that they will continue in the future. Un grand merci également à la région Bretagne d'avoir partiellement financé ce déplacement.

Je souhaite aussi remercier tous les permanents, post-doctorants, doctorants et stagiaires de l'IETR avec qui j'ai pu échanger au cours des 3 (voire 5) dernières années. Un grand merci à Imane Assoum, Zachary Khodja, Elias Naciri, Maëlia Moreno et Erwan Jamet pour le travail fourni lors de leurs stages respectifs. Ces stages ont permis de mûrir la réflexion et d'enrichir certaines parties des travaux présentés dans cette thèse.

Un grand merci à Mme Gauthier, ma professeure de mathématiques au lycée, pour m'avoir fait prendre goût à la beauté, la rigueur et l'abstraction des mathématiques. Vous avez joué un rôle très important quant à mon choix de poursuivre vers des études théoriques, et avez, à votre échelle, orienté mon parcours vers cette thèse.

Un grand merci à mes différents co-bureaux de thèse. Corentin, merci pour ton expérience, pour les conseils que tu as pu me fournir, et ta patience devant mon envie de rester dans le noir. Clément, merci pour ta gentillesse et pour m'avoir fait découvrir un autre monde que le traitement du signal. Taha (grand frère de thèse) merci de m'avoir si bien accompagné pendant mon stage de fin d'études ainsi que pendant le début de la thèse. Nay (petite sœur de thèse) merci pour ton humour et ta présence au quotidien, et merci de m'avoir laissé gagner les batailles de lumières (encore). Je suis sûr et certain que ta thèse se passera de manière formidable.

Un grand merci à tous mes amis pour leur soutien. Merci à mes amis de l'INSA Rennes pour les années passées ensemble. Tout particulièrement, merci à Marion, Lou, Camille et Sophie pour leur amitié depuis bientôt 10 ans, même si nous sommes maintenant éparpillés au-delà des frontières bretonnes. Pour les amis rencontrés en chemin, merci à Thomas pour nos précieuses escapades cyclistes, à Nay (encore) pour tous tes conseils de design de diapos, mais également de vie (et aussi pour la découverte du chich taouk), à Maxime et Émile pour les différents fous rires le midi et à Lucas pour les parties de Dune : Imperium. Un merci particulier à Corentin (à nouveau) et à Lisa pour leurs amitiés qui me sont chères. Corentin, merci pour les découvertes hétéroclites : des chemins du Queyras, à Projet Gaïa et à l'optimisation des BOs, ces dernières années ont été très variées ! Lisa, merci d'avoir contribué à ma consommation excessive d'espresso à des heures (trop) matinales, mais aussi pour ton écoute et ton soutien. Tu as gagné, je reconnais officiellement que l'Italie a une cuisine qui ne se résume pas qu'aux pâtes.

Un grand merci à mes beaux-parents pour leur accueil, leur gentillesse ainsi que leur présence à la soutenance.

Un énorme merci à mes parents et mon petit frère pour leur amour, leurs conseils ainsi que leur soutien inconditionnel au cours de toutes ces années. Merci de m'avoir toujours aidé à faire les meilleurs choix possibles, quels que soient les nouveaux projets farfelus que je pouvais vous présenter. Un grand merci à l'ensemble de ma famille, notamment mes grands-parents, que j'espère rendre fiers à travers ces travaux. Mamie Christiane, tu n'auras malheureusement pas eu la chance de voir la fin de cette thèse, mais je suis sûr que tu me fais des boujoux de là où tu es !

Finalement, je ne pourrais finaliser ces remerciements sans mentionner celle qui partage ma vie. Marion, merci pour tout et encore plus. Merci de bousculer un peu mes habitudes et de me faire découvrir de nouveaux horizons culturels (promis j'aime Depardon pour de vrai). Si ces quelques derniers mois ont été vraiment intenses, ta présence les a rendus plus supportables. Merci de supporter mes heures sur le vélo et au piano. Bref, merci d'être toi, et j'espère que ces quelques mots sauront retranscrire tout mon amour.

10/02/2026,

Baptiste Chatelier

Abstract

The field of wireless communications has traditionally relied on signal processing techniques, based on mathematical models, to describe and optimize wireless transmission of information. However, when the underlying assumptions of these models are not fully satisfied, conventional methods often exhibit suboptimal performance. In recent years, machine learning techniques have emerged as a promising alternative to address these limitations, owing to their adaptability and data-driven nature. Nevertheless, machine learning-based approaches also entail additional challenges, particularly in terms of interpretability and computational complexity.

This thesis investigates the use of the model-based machine learning paradigm in wireless communication or sensing systems. This paradigm seeks to combine the strengths of both model-based and machine learning approaches, while mitigating their respective weaknesses. In particular, it leverages the mathematical models used in model-based approaches to structure, initialize, or optimize learning methods.

Three application areas are explored within this thesis. First, it is proposed to use a physical propagation channel model to structure a neural architecture that learns the location-to-channel mapping. This neural network is subsequently used to improve the performance of traditional radio-localization techniques. Second, as practical wireless systems are inherently affected by hardware impairments, it is proposed to apply the model-based machine learning paradigm to mitigate their impact. Third, the growing number of antennas in modern cellular networks leads to a rapid increase in the dimensionality of the channel, thereby significantly increasing computational complexity. To mitigate this issue, this thesis investigates the use of a model-based dimensionality reduction technique known as channel charting.

Table of contents

Acknowledgements i

Abstract v

Table of contents vii

Acronyms xi

Mathematical notations xv

Synthèse de la thèse en français xvii
Contexte de la thèse . . . xvii
Synthèse des contributions . . . xx
Liste des publications . . . xxiv
Mobilité internationale . . . xxv

Introduction 1
Context . . . 1
Thesis outline . . . 5
List of publications . . . 8
International mobility . . . 9

I Background and preliminary concepts 11

1 Propagation channel 13
1.1 Propagation channel and physical phenomena . . . 14
1.2 Mathematical model of the propagation channel . . . 20
1.3 Application to modern communication systems . . . 31
1.4 Conclusion . . . 45

2 Machine learning 47
2.1 Typical problems . . . 48
2.2 General learning notions . . . 56
2.3 Some popular ML methods . . . 64

2.4 Focus on deep learning . . . 68
2.5 Conclusion . . . 78

3 Model-based machine learning **79**
3.1 General context . . . 80
3.2 Towards model-based machine learning . . . 90
3.3 Application to the wireless channel . . . 99
3.4 Conclusion . . . 108

II Contributions **109**

4 Location-to-channel mapping: learning **111**
4.1 Problem setting . . . 112
4.2 Towards a model-based approach . . . 119
4.3 Model-based neural architecture . . . 125
4.4 Experiments . . . 128
4.5 Conclusion . . . 140

5 Location-to-channel mapping: localization application **143**
5.1 Introduction . . . 144
5.2 Proposed approach . . . 148
5.3 Experiments . . . 160
5.4 Conclusion . . . 170

6 Channel estimation and impairments learning **173**
6.1 Introduction . . . 174
6.2 Model-based approach . . . 175
6.3 Proposed approach . . . 178
6.4 Experimental results . . . 183
6.5 Conclusion . . . 188

7 DoA estimation and impairments learning **189**
7.1 Problem setting . . . 190
7.2 Model-based approach . . . 192
7.3 Towards diffMUSIC . . . 194
7.4 Experiments . . . 198
7.5 Conclusion . . . 204

8 Channel charting and beam prediction **205**
8.1 Primer on channel charting . . . 206
8.2 Problem setting . . . 211
8.3 Proposed method . . . 215
8.4 Experiments . . . 218

8.5 Conclusion 224

9 Channel charting and CSI compression **225**
9.1 Problem setting 226
9.2 Proposed approach 230
9.3 Experiments 233
9.4 Conclusion 238

Conclusion and Perspectives **241**
Thesis conclusions 241
Perspectives 243

III Appendices and bibliography **247**

A Chapter 1's appendix **249**
A.1 Proof of Eq. (1.2) 249
A.2 Proof of Definition 1.2 250
A.3 Proof of Eq. (1.30) 250
A.4 Proof of η formulation in Eq. (1.53) 251

B Chapter 2's appendix **253**
B.1 Proof of Eq. (2.27) 253
B.2 Proof of Eq. (2.33) 254

C Chapter 3's appendix **257**
C.1 Proof of Eq. (3.10) 257

D Chapter 4's appendix **259**
D.1 Proof of Lemma 4.1 259
D.2 Proof of Corollary 4.1 259
D.3 Proof of Proposition 4.1 261
D.4 Proof of Theorem 4.2 262

E Chapter 5's appendix **265**
E.1 Proof of Theorem 5.1 265
E.2 Proof of Theorem 5.2 265
E.3 Proof of Theorem 5.3 267
E.4 Proof of Corollary 5.1 268
E.5 Proof of Corollary 5.2 268

F Chapter 7's appendix **271**
F.1 About the subspace separation of $\mathbf{\Gamma_X}$ 271
F.2 $\mu_{\mathrm{SL},P}$ and subspace orthogonalization 273

Bibliography **277**

Acronyms

k-NN	k-Nearest Neighbors
AE	Autoencoder
AI	Artificial Intelligence
AWGN	Average White Gaussian Noise
bps	Bit Per Second
BS	Base Station
CC	Channel Charting
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CF	Cell Free
CIR	Channel Impulse Response
CNN	Convolutional Neural Network
CSI	Channel State Information
CT	Continuity
DFT	Discrete Fourier Transform
DoA	Direction of Arrival
DoD	Direction of Departure
EGT	Equal Gain Transmission
EVD	Eigenvalue Decomposition
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FRV	Frequency Response Vector
GARF	Gaussian Activated Radiance Fields
GPU	Graphical Processing Unit
GT	Ground Truth

HPBW	Half Power Beam Width
HWI	Hardware Impairements
INR	Implicit Neural Representation
ISAC	Integrated Sensing And Communications
KAN	Kolmogorov-Arnold Networks
KS	Kruskal's stress
LLM	Large Language Model
LoS	Line-of-Sight
LRA	Low Rank Approximation
LS	Least Squares
MB-ML	Model Based Machine Learning
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MLP	Multi Layer Perceptron
mMIMO	Massive Multiple Input Multiple Output
MMSE	Minimum Mean Squared Error
mmWave	millimeter Wave
MP	Matching Pursuit
MRT	Maximum Ratio Transmission
MSE	Mean Squared Error
MU-MIMO	Multiple User Multiple Input Multiple Output
MUSIC	MUltiple SIgnal Classification
NLoS	Non Line-of-Sight
NMSE	Normalized Mean Squared Error
NN-aided	Neural Network-aided
NTK	Neural Tangent Kernel
OMP	Orthogonal Matching Pursuit

PCA	Principal Component Analysis
PDE	Partial Differential Equation
PI	Phase Insensitive
PINN	Physics-Informed Neural Network
PS	Phase Sensitive
RFFs	Random Fourier Features
RKHS	Reproducing Kernel Hilbert Space
RMSPE	Root Mean Squared Periodic Error
SCO	Sampling Clock Offset
SDMA	Spatial Division Multiple Access
SGD	Stochastic Gradient Descent
SINR	Signal-to-Interference-and-Noise Ratio
SIR	Signal-to-Interference Ratio
SIREN	SInusoidal REpresentation Network
SISO	Singe Input Single Input
SL	Supervised Learning
SNR	Signal-to-Noise Ratio
SR	Sum Rate
SV	Steering Vector
SVD	Singular Value Decomposition
TDMA	Time Division Multiple Access
TW	Trustworthiness
UE	User Equipment
UL	Unsupervised Learning
ULA	Uniform Linear Array
UPA	Uniform Planar Array
ZF	Zero Forcing

MATHEMATICAL NOTATIONS

The main mathematical notations used throughout this thesis are summarized in Table 1.

TABLE 1 : Notational conventions.

Notation	Description
f, g	Arbitrary functions
x, θ	Arbitrary scalars
$\mathbf{x}, \boldsymbol{\theta}$	Arbitrary vectors
$\mathbf{X}, \boldsymbol{\Theta}$	Arbitrary matrices
$\mathcal{X}, \mathcal{Y}$	Arbitrary sets
$\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{R}^+, \mathbb{C}$	Sets of natural, integer, real, positive real, and complex numbers
$\mathbb{R}^n, \mathbb{C}^n$	Sets of real and complex vectors of dimensions n
$\mathbb{R}^{n\times m}, \mathbb{C}^{n\times m}$	Sets of real and complex matrices of dimensions $n \times m$
$\varnothing$	Empty set
$\mathcal{Y}^{\mathcal{X}}$	Set of all functions from $\mathcal{X}$ to $\mathcal{Y}$
$C(\mathcal{X}, \mathcal{Y})$	Set of continuous functions from $\mathcal{X}$ to $\mathcal{Y}$
$\mathrm{SO}(n)$	nth special orthogonal group,
	i.e. $\mathrm{SO}(n) = \left\{\mathbf{R} \in \mathbb{R}^{n\times n} \middle\vert \mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{Id}_n, \det(\mathbf{R}) = 1\right\}$
$\mathcal{C}(\mathbf{c}, r)$	Circle of center $\mathbf{c} \in \mathbb{R}^2$, and radius $r \in \mathbb{R}^+$,
	i.e. $\forall \mathbf{c} \in \mathbb{R}^2, r \in \mathbb{R}^+, \mathcal{C}(\mathbf{c}, r) = \{\mathbf{x} \in \mathbb{R}^2 \vert \, \lVert\mathbf{x} - \mathbf{c}\rVert_2 = r\}$
$\mathcal{S}(\mathbf{c}, r)$	Sphere of center $\mathbf{c} \in \mathbb{R}^3$, and radius $r \in \mathbb{R}^+$,
	i.e. $\forall \mathbf{c} \in \mathbb{R}^3, r \in \mathbb{R}^+, \mathcal{S}(\mathbf{c}, r) = \{\mathbf{x} \in \mathbb{R}^3 \vert \, \lVert\mathbf{x} - \mathbf{c}\rVert_2 = r\}$
$\dim_{\mathcal{X}}(\mathcal{Y})$	Dimension of $\mathcal{Y}$ in the space $\mathcal{X}$
$\lvert\mathcal{X}\rvert$	Cardinal of $\mathcal{X}$
$\mathcal{X} \times \mathcal{Y}$	Cartesian product of $\mathcal{X}$ and $\mathcal{Y}$
$\mathcal{X} \cap \mathcal{Y}$	Intersection of $\mathcal{X}$ and $\mathcal{Y}$
$\mathcal{X} \cup \mathcal{Y}$	Union of $\mathcal{X}$ and $\mathcal{Y}$
$\mathrm{dom}(f)$	Domain of f,
	i.e. $f\colon \mathcal{X} \to \mathcal{Y} \Rightarrow \mathrm{dom}(f) = \mathcal{X}$
$\mathrm{dom}^{\mathsf{c}}(f)$	Codomain of f,
	i.e. $f\colon \mathcal{X} \to \mathcal{Y} \Rightarrow \mathrm{dom}^{\mathsf{c}}(f) = \mathcal{Y}$
$\lvert f(x)\rvert$	Absolute value of $f(x)$ if $f(x) \in \mathbb{R}$, modulus if $f(x) \in \mathbb{C}$
$\langle f(x), g(x)\rangle_{\mathcal{Y}}$	Scalar product of $\mathcal{Y}$ between $f(x) \in \mathcal{Y}$ and $g(x) \in \mathcal{Y}$

$f(x) \circledast g(x)$	Convolution of $f(x)$ with $g(x)$
$\mathcal{F}_x[f(x,y)]$	Fourier transform of $f(x,y)$ on the x variable
$\mathbb{E}_{x\sim p_x}[f(x,y)]$	Expectation of $f(x,y)$, over the x random variable, following distribution p_x
$\mathfrak{Re}(f(x)), \mathfrak{Im}(f(x))$	Real, Imaginary parts of $f(x) \in \mathbb{C}$
$f(x) = \mathcal{O}(g(x))$	Big-O Bachman-Landau notation i.e. $\exists k > 0, \exists x_0, \forall x \geq x_0, \lvert f(x)\rvert \leq kg(x)$
$f(x) = o(g(x))$	Small-o Bachman-Landau notation i.e. $\forall k > 0, \exists x_0, \forall x \geq x_0, \lvert f(x)\rvert \leq kg(x)$
$\nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y})$	Gradient of $f(\mathbf{x},\mathbf{y})$ wrt. $\mathbf{x}$ i.e. $\mathbf{x} \in \mathbb{R}^N$, $\nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y}) = \left[\frac{\partial}{\partial x_1} f(\mathbf{x},\mathbf{y}), \cdots, \frac{\partial}{\partial x_N} f(\mathbf{x},\mathbf{y})\right]$
$\Delta_{\mathbf{y},\mathbf{x}}$	Jacobian matrix of $f\colon \mathbb{R}^n \to \mathbb{R}^m$, s.t. $\mathbf{y} = f(\mathbf{x})$ i.e. $\Delta_{\mathbf{y},\mathbf{x}} = [\nabla_{x_1}\mathbf{y}, \cdots, \nabla_{x_n}\mathbf{y}] \in \mathbb{R}^{m\times n}$
$\mathbb{1}_{\mathcal{X}}(x)$	Indicator function on $\mathcal{X}$ i.e. $\mathbb{1}_{\mathcal{X}}(x) = 1$ if $x \in \mathcal{X}$, $\mathbb{1}_{\mathcal{X}}(x) = 0$ if $x \notin \mathcal{X}$
$\delta(x-a)$	Dirac distribution centered on a i.e. $\delta(x-a) = 1$ if $x = a$, $\delta(x-a) = 0$ if $x \neq a$
$\lVert\mathbf{x}\rVert_p$	ℓ_p norm of $\mathbf{x}$, $p \in \mathbb{N}^*$
$\lVert\mathbf{x}\rVert_0$	ℓ_0 norm of $\mathbf{x}$
$\lVert\mathbf{X}\rVert_{\mathsf{F}}$	Frobenius norm of $\mathbf{X}$
$\mathrm{diag}(\mathbf{x})$	Diagonal matrix constructed from $\mathbf{x}$
$\mathrm{rk}(\mathbf{X})$	Rank of $\mathbf{X}$
$\mathrm{Tr}(\mathbf{X})$	Trace of $\mathbf{X}$
$\det(\mathbf{X})$	Determinant of $\mathbf{X}$
$\mathrm{span}(\mathbf{X})$	Linear span of $\mathbf{X}$
$\mathrm{vec}(\mathbf{X})$	Vectorization of $\mathbf{X}$
$\mathbf{X}^*$	Point-wise conjugate of $\mathbf{X}$
$\mathbf{X}^{\mathsf{T}}$	Transpose of $\mathbf{X}$
$\mathbf{X}^{\mathsf{H}}$	Hermitian of $\mathbf{X}$
$\mathbf{X} \otimes \mathbf{Y}$	Kronecker product between $\mathbf{X}$ and $\mathbf{Y}$
$\mathbf{X} \odot \mathbf{Y}$	Hadamard product between $\mathbf{X}$ and $\mathbf{Y}$
$\mathbf{Id}_d$	Identity matrix of dimension $d \times d$.
$\mathcal{U}(\mathcal{X})$	Uniform distribution on the subspace defined by $\mathcal{X}$
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	d-dimensional gaussian distribution, with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$
$\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	d-dimensional complex gaussian distribution, with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$

Synthèse de la thèse en français

Note pour le lecteur / Note to the reader

Cette section propose un résumé étendu de la thèse, rédigé en français. Le corps principal du manuscrit, rédigé en anglais, commence à la suite de cette section.

This section offers an extended summary of the thesis in French. The main body of the manuscript, written in English, begins thereafter.

Contexte de la thèse

Depuis l'apparition des premières formes de l'espèce humaine, les êtres humains ont développé des méthodes de plus en plus complexes pour communiquer entre eux. Ce processus a commencé avec l'acquisition du langage par l'*Homo sapiens*, qui a jeté les bases de l'émergence de langues structurées. Les modes de communication visuels ont suivi, d'abord sous la forme d'art pariétal, puis avec le développement des premiers systèmes d'écriture tels que le cunéiforme et les hiéroglyphes. Ces systèmes ont permis aux humains d'enregistrer, de stocker et de transmettre des informations à travers le temps et l'espace. L'introduction des alphabets a ensuite conduit à la stabilisation des langues au sein des régions, contribuant à l'augmentation du taux d'alphabétisation, à l'accès à l'éducation et à une plus grande efficacité dans toutes les formes d'activités sociales, de l'agriculture à l'art et à la médecine. Un tournant majeur a été marqué par l'invention de l'imprimerie au XVe siècle, une innovation révolutionnaire qui a permis la production massive de livres et qui a ainsi considérablement élargi la diffusion des connaissances. Cependant, les livres en tant que moyen de communication présentent plusieurs limites : ils exigent la maîtrise de la lecture, ne sont pas universellement accessibles et permettent une diffusion et une mise à jour relativement lente des informations.

La transmission rapide d'informations sur de longues distances a commencé dès le VIIe siècle avant notre ère, avec des systèmes rudimentaires tels que les signaux de fumée et les balises lumineuses. Les systèmes modernes de transmission à longue distance remontent au télégraphe optique de Claude Chappe, mis au point en 1790. Ce système mécanique rudimentaire a considérablement réduit les délais de transmission de l'information : par exemple, la transmission d'un message entre Paris et Strasbourg, deux villes françaises distantes d'environ 500 km, est passée de quatre jours à cheval à seulement deux heures grâce au télégraphe de Chappe. Les progrès scientifiques et technologiques rapides qui ont suivi ont conduit au développement de la télégraphie électromagnétique, avec les contributions fondamentales de Samuel Thomas von Sömmerring (1809), Francis Ronalds (1816), André-Marie Ampère (1820), William Fothergill Cook et Charles Wheatstone

(1837), et Samuel Morse (1838). En 1876, Alexander Graham Bell inventa le téléphone, permettant ainsi le premier appel transcontinental dès la fin de l'année 1914. Le télégraphe et le téléphone étaient tous deux des moyens de *communication filaires*, car ils reposaient sur des fils électriques pour transmettre l'information. Par conséquent, ces technologies étaient confrontées à des limites fondamentales, notamment liées aux coûts de déploiement et de maintenance élevés, ainsi qu'une vulnérabilité aux pannes liée à la relative fragilité du moyen de transmission.

Dans les années 1890, Guglielmo Marconi a été le pionnier des *communications sans fil* en combinant la technologie des ondes radio, théorisée par James Clerk Maxwell (1865) et confirmée expérimentalement par Heinrich Hertz (1887), avec les principes télégraphiques, ce qui a conduit au développement du premier système de télégraphie sans fil. Cette innovation a finalement évolué pour devenir ce que l'on considère aujourd'hui comme étant les systèmes radio. Les deux guerres mondiales et la guerre froide qui a suivi au XXe siècle ont déclenché une course effrénée à l'innovation technologique. Cette période a été marquée par des avancées majeures dans le domaine des communications sans fil, avec des innovations clés telles que la détection et la localisation par radio (RADAR), grâce aux contributions majeures de Robert Watson Watt et Hans Hollmann dans les années 1930, ou le développement des communications par satellite et de localisation par satellite (GPS) dans les années 1970. Les années 1980 ont marqué le début de l'ère des réseaux cellulaires sans fil. La première génération (1G), vers 1980, ne prenait en charge que l'audio analogique. La deuxième génération (2G), vers 1990, a introduit la prise en charge des signaux numériques et de l'envoi de messages textuels. La troisième génération (3G), vers 2000, a augmenté les débits de transmission de données et a introduit l'accès mobile à Internet, permettant aux utilisateurs de se connecter à Internet via des appareils portables. La quatrième génération (4G), vers 2010, a encore augmenté les débits de données, permettant la transmission vidéo sans fil en temps réel. La cinquième génération (5G), vers 2020, a apporté des débits de transmission nettement plus élevés, une bande passante accrue et des capacités de communication à faible latence et haute fiabilité. La sixième génération (6G), attendue vers 2030, vise à encore améliorer les capacités de transmission de données. Outre l'amélioration de la vitesse et de la bande passante, la 6G pourrait également intégrer des considérations de soutenabilité environnementale dans sa conception, et adopter l'intelligence artificielle/l'apprentissage automatique (IA/ML) comme composante essentielle de ses futures fonctionnalités.

La dernière décennie a été marquée par un essor considérable de l'application des techniques d'IA/ML dans un large éventail de problèmes. Cette tendance est illustrée par le prix Nobel de physique 2024, décerné à John Hopfield et Geoffrey Hinton en reconnaissance de leurs contributions pionnières aux réseaux de neurones artificiels. Des domaines tels que l'industrie automobile, la santé, la finance ou les télécommunications ont de plus en plus intégré les méthodes d'apprentissage automatique dans leurs pratiques courantes. Cette adoption ne se limite pas au milieu professionnel : le grand public s'est récemment approprié les méthodes d'IA à travers des agents conversationnels tels que ChatGPT, LeChat ou Claude, ou encore des outils de génération d'images comme DALL-E ou Mid-

journey. Cet intérêt croissant peut être attribué à plusieurs facteurs. Tout d'abord, les progrès technologiques réalisés dans le domaine de la microélectronique au cours des trente dernières années ont permis d'accéder à des ressources informatiques efficaces et puissantes, ce qui a permis le développement de méthodes d'IA/ML de plus en plus complexes. En effet, si le domaine de l'apprentissage automatique remonte aux années 1960, les progrès réalisés à l'époque étaient limités par le manque de ressources informatiques suffisamment puissantes. Deuxièmement, par définition, les méthodes d'apprentissage automatique nécessitent une phase d'entraînement, qui peut exiger des ensembles de données de grande taille. Si l'humanité a toujours généré des connaissances et des informations, le développement et l'expansion rapide d'Internet au cours des trois dernières décennies ont considérablement augmenté la quantité de données disponibles. On estime par exemple qu'en 2025, le volume de données générées quotidiennement est d'environ 400 milliards de gigaoctets [Des19]. Ensemble, ces évolutions ont considérablement amélioré les performances, la facilité d'utilisation et l'impact des méthodes d'apprentissage automatique, tant dans des contextes spécialisés que dans la vie quotidienne.

Malgré cette croissance rapide et leur utilisation croissante pour résoudre un large éventail de problèmes, les méthodes d'apprentissage automatique se heurtent encore à plusieurs limites importantes. Premièrement, les modèles modernes se caractérisent par une grande complexité algorithmique. Par exemple, les grands modèles de langage (LLM) utilisés dans les agents conversationnels peuvent contenir jusqu'à des milliers de milliards de paramètres, qui doivent tous être optimisés pendant une phase d'entraînement. Il en résulte des temps d'entraînement extrêmement longs et des coûts associés colossaux : à titre d'exemple, les LLMs modernes nécessitent plusieurs mois d'apprentissage sur des architectures informatiques distribuées complexes, avec des coûts associés pouvant atteindre plusieurs millions de dollars américains. Deuxièmement, ces modèles exigent des jeux de données massifs pour atteindre leurs performances. Malgré la croissance exponentielle du volume de données disponibles observée ces dernières années, certains travaux de recherche suggèrent que l'ensemble des données textuelles produites par l'humanité et accessibles publiquement pourrait être épuisé pour l'entraînement des LLMs d'ici 2030. Cette projection suscite des interrogations majeures concernant la pérennité et l'évolutivité des avancées futures dans le domaine des modèles de langage. Troisièmement, en raison de leur extrême complexité, les modèles modernes souffrent d'une interprétabilité limitée. Ils sont donc souvent qualifiés de *boîtes noires* : dans de nombreux cas, il est fondamentalement difficile de comprendre pourquoi une entrée donnée conduit à une sortie particulière.

Ces dernières années, une attention particulière a été accordée à l'amélioration de l'interprétabilité des modèles d'apprentissage automatique, dans le but d'atténuer leur nature "boite noire". L'une des principales orientations de recherche dans ce domaine est le paradigme dit *d'apprentissage automatique basé sur des modèles*. Ce paradigme vise à combiner la rigueur et la structure des modèles mathématiques traditionnellement utilisés dans le traitement du signal avec la flexibilité et les capacités d'apprentissage des méthodes modernes d'apprentissage automatique. En effet, dans le domaine du traitement

du signal, les solutions proposées s'appuient généralement sur des modèles mathématiques simples, soigneusement élaborés à partir des connaissances a priori sur le problème étudié. Ces connaissances a priori proviennent généralement d'une compréhension des principes physiques qui régissent le système en question, fournissant ainsi une base interprétable pour la solution obtenue. En raison de leur dépendance à un modèle mathématique donné, ces approches sont qualifiées de *basées sur des modèles*. Cependant, comme l'a observé George E. P. Box, *"Tous les modèles sont faux, mais certains sont utiles"* : les modèles mathématiques utilisés dans le traitement du signal sont généralement construits sur des hypothèses simplificatrices, ce qui limite intrinsèquement leur validité. Par conséquent, même des déviations mineures par rapport au système initialement étudié peuvent conduire à des solutions dérivées de modèles simplifiés qui sont sous-optimales. En revanche, les méthodes d'apprentissage automatique sont dites *basées sur les données*, car elles apprennent des solutions directement à partir des données fournies aux méthodes en question. Cela leur confère une flexibilité et une robustesse accrues face aux déviations mineures par rapport au système supposé. Cependant, comme mentionné précédemment, cette flexibilité se fait généralement au détriment de l'interprétabilité. Le paradigme de l'apprentissage automatique basé sur des modèles vise donc à combiner les points forts des deux approches : tirer parti de l'adaptabilité des méthodes basées sur les données, tout en conservant l'interprétabilité et la structure fournies par les méthodes basées sur des modèles.

Les communications sans fil nécessitent des méthodes de traitement du signal qui s'appuient sur des modèles mathématiques pour décrire et optimiser la transmission d'informations sans fil. Si les méthodes d'apprentissage automatique ont récemment été utilisées pour améliorer certaines techniques de communication, elles soulèvent également d'importantes questions en matière d'interprétabilité et de complexité. Compte tenu à la fois de l'abondance des modèles mathématiques établis, et des limites des approches purement basées sur les données, le domaine des communications sans fil semble particulièrement bien adapté à l'utilisation du paradigme de l'apprentissage automatique basé sur des modèles. Cette observation motive la question centrale de cette thèse, telle que présentée ci-dessous.

Problématique principale de la thèse

Le paradigme de l'apprentissage automatique basé sur des modèles peut-il être utilisé pour fournir des solutions interprétables et à complexité limitée pour des problèmes complexes de communication sans fil ?

Synthèse des contributions

La question posée précédemment est intrinsèquement vaste, et y répondre de manière exhaustive nécessiterait l'évaluation des méthodes d'apprentissage automatique basées

sur des modèles sur l'ensemble des problèmes liés aux communications sans fil, une tâche dépassant le cadre d'une seule thèse. Par conséquent, cette thèse se concentre sur trois axes de recherche principaux, présentés ci-dessous.

Axes de recherche

(A1) **Fonction position-canal.** Est-il possible de concevoir une architecture neuronale qui apprend la fonction liant une position au canal de propagation associé ? Si oui, est-il possible d'utiliser cette architecture neuronale pour améliorer la précision des méthodes de radiolocalisation ?

(A2) **Compensation des imperfections matérielles.** Dans les systèmes de communication sans fil, les imperfections matérielles dégradent significativement les performances de transmission. Dans ce contexte, est-il possible d'utiliser des méthodes d'apprentissage automatique basées sur des modèles pour compenser l'impact de ces imperfections ?

(A3) **Compression de canal.** Dans les réseaux cellulaires modernes, la grande dimension des coefficients des canaux de propagation induit une complexité importante. La cartographie de canal, une méthode de réduction de dimension basée sur un modèle, peut-elle être utilisée pour réduire cette complexité ?

L'objectif de cette thèse est de faire progresser l'état actuel des connaissances dans les trois axes de recherche identifiés. Ce manuscrit est structuré en deux parties, et organisé en neuf chapitres, comme indiqué ci-dessous. Chaque chapitre peut être lu indépendamment des autres. Cependant, l'auteur recommande fortement au lecteur de consulter le chapitre 3, qui fournit une introduction détaillée aux concepts fondamentaux de cette thèse.

Partie I : Contexte et concepts préliminaires

Chapitre 1

Ce chapitre présente les concepts fondamentaux liés au canal de propagation. Il montre notamment comment un modèle mathématique du canal de propagation peut être dérivé à partir des principes physiques décrivant les interactions entre les ondes et la matière. Ce chapitre présente ensuite plusieurs exemples illustrant comment les techniques de communication modernes exploitent la connaissance du canal de propagation pour améliorer les performances de transmission.

Chapitre 2

Ce chapitre fournit une introduction générale au domaine de l'apprentissage automatique. Après avoir présenté les différents types de problèmes d'apprentissage et les notions fondamentales de l'apprentissage, ce chapitre présente en détail trois méthodes usuelles

d'apprentissage automatique : la régression linéaire, la méthode des k plus proches voisins, et l'analyse en composantes principales. Un intérêt particulier est ensuite porté à l'apprentissage profond, car il constitue le cœur de la plupart des méthodes modernes d'apprentissage automatique. À ce titre, les notions théoriques générales concernant les réseaux de neurones et les procédures d'entraînement associées sont présentées. Enfin, ce chapitre aborde les limites actuellement connues des réseaux de neurones profonds.

Chapitre 3

Ce chapitre présente en détail le paradigme de l'apprentissage automatique basé sur les modèles. Il souligne en particulier comment cette approche combine les forces des approches basées sur les modèles et de celles basées sur l'apprentissage automatique, améliorant ainsi efficacement l'interprétabilité des modèles et palliant le manque de connaissances a priori. Ce chapitre présente ensuite comment ce paradigme peut être appliqué aux problèmes décrits dans les axes de recherche (A1)-(A3), et sert ainsi d'introduction aux contributions de cette thèse.

Partie II : Contributions

Chapitre 4

Ce chapitre se concentre sur (A1) : la conception d'un réseau neuronal basé sur un modèle qui apprend la fonction position-canal est présentée. Cette contribution s'appuie sur le concept de représentations neuronales implicites, où les réseaux de neurones sont utilisés pour approximer des fonctions continues. Le réseau de neurones proposé est directement obtenu à la suite de développements mathématiques sur un modèle de canal de propagation basé sur la physique, ce qui permet d'obtenir une structure interprétable et des performances d'apprentissage explicables. Les résultats expérimentaux présentés montrent que le modèle proposé apprend avec succès la fonction souhaitée, et que, contrairement aux approches neuronales classiques, il surmonte le problème du biais spectral. En effet, le réseau de neurones présenté est capable de capturer à la fois les composantes basse et haute fréquence de la fonction position-canal, ce qui peut s'expliquer par le biais de sa structure.

Chapitre 5

Ce chapitre s'intéresse à (A1) en appliquant le réseau de neurones proposé au chapitre 4 au problème de la localisation sans fil. En particulier, le réseau est utilisé en tant que modèle neuronal génératif de canal, afin d'améliorer la méthode classique de localisation dite de *fingerprinting*. En tirant parti de sa capacité à prédire les coefficients de canal à des positions arbitraires, le réseau enrichit la base de données de fingerprinting, améliorant ainsi la précision de localisation. En outre, des développements théoriques sur la mesure de similarité utilisée pour effectuer le fingerprinting sont exploités afin d'améliorer encore la précision de la méthode de localisation proposée. Des expériences approfondies montrent

que la méthode proposée atteint une précision de localisation inférieure à la longueur d'onde, même dans des scénarios difficiles sans ligne de vue directe.

Chapitre 6

Ce chapitre se concentre sur (A2) en améliorant un réseau de neurones basé sur un modèle, conçu pour réaliser une tâche d'estimation du canal et d'apprentissage d'imperfections matérielles. Le réseau déjà existant est utilisé dans un système de communication sans fil multifréquence pour apprendre des imperfections de gains d'antenne ainsi que des décalages de fréquence. Le nombre de paramètres d'apprentissage et la complexité temporelle de ce réseau sont optimisés en suivant le paradigme de l'apprentissage automatique basé sur un modèle. Les résultats expérimentaux attestent de l'efficacité de l'approche proposée, tant pour l'apprentissage des imperfections matérielles que pour l'estimation précise des canaux.

Chapitre 7

Ce chapitre développe (A2) en introduisant un algorithme différentiable qui, dans un système multi-antenne, effectue conjointement l'estimation de la direction d'arrivée des ondes incidentes, et l'apprentissage des imperfections matérielles du système. L'approche proposée s'appuie sur l'algorithme MUSIC, en le modifiant pour le rendre différentiable. De plus, des fonctions de coût supervisées et non supervisées adaptées à la tâche considérée sont introduites pour améliorer les performances d'apprentissage. Les résultats expérimentaux montrent que la méthode proposée surpasse l'algorithme MUSIC non différentiable en termes de performances.

Chapitre 8

Ce chapitre est consacré à (A3), dans le but d'utiliser la cartographie de canal, une méthode de réduction de dimension non supervisée, pour réduire la complexité de la gestion des faisceaux dans les systèmes cell-free massive MIMO. En utilisant la méthode de cartographie de canal, les coefficients de canaux sont compressés en une représentation de faible dimension, qui est ensuite partagée aux différentes stations de base du système. Au niveau de chaque station de base, un réseau de neurones adapté est utilisé pour décoder l'information de faible dimension en un indice d'un dictionnaire de faisceaux prédéfini, permettant ainsi une formation de faisceaux efficace. Les résultats d'expériences montrent l'efficacité de l'approche proposée, et mettent en évidence ses avantages en termes de performance par rapport à des méthodes de référence.

Chapitre 9

Ce chapitre se focalise sur (A3) dans le but d'utiliser la cartographie de canal pour proposer une méthode de compression des canaux spécifique à une tache de formation de faisceaux pour les systèmes massive MIMO. Cette contribution consiste en la compression

des coefficients de canaux au niveau des utilisateurs grâce à la méthode de cartographie de canal, la transmission de l'information de faible dimension qui en résulte à la station de base, puis son décodage en une matrice de précodage à l'aide d'un réseau de neurones adapté. Une stratégie d'apprentissage heuristique est proposée, et les performances de l'approche présentée sont explorées avec différents taux de compression, démontrant des performances supérieures par rapport aux méthodes classiques.

Dans la Partie II, le coût de calcul associé à chaque contribution est systématiquement indiqué en heures GPU, au début de chaque chapitre. Les calculs ayant été faits principalement sur un cluster de GPU NVIDIA A40, les temps rapportés correspondent à un temps de calcul équivalent sur un unique GPU NVIDIA A40.

Liste des publications

Le travail présenté dans ce manuscrit est basé sur les publications suivantes.

Articles de revue

(J1) Baptiste Chatelier, Vincent Corlay, Musa Furkan Keskin, Matthieu Crussière, Henk Wymeersch et Luc Le Magoarou, "Model-based Implicit Neural Representation for sub-wavelength Radio Localization", *soumis à IEEE Transactions of Wireless Communications*, Juin 2025 ; actuellement en révision

(J2) Baptiste Chatelier, Vincent Corlay, Matthieu Crussière et Luc Le Magoarou, "Model-Based Learning for Multi-Antenna Multi-Frequency Location-to-Channel Mapping", *à IEEE Journal of Selected Topics in Signal Processing*, vol. 19, no. 3, pp. 520-535, Avril 2025

Articles de conférence

(C1) Baptiste Chatelier, José Miguel Mateos-Ramos, Vincent Corlay, Christian Häger, Matthieu Crussière, Henk Wymeersch et Luc Le Magoarou, "Physically Parameterized Differentiable MUSIC for DoA Estimation with Uncalibrated Arrays", *à IEEE International Conference on Communications*, Montréal, Canada, 2025

(C2) Baptiste Chatelier, Vincent Corlay, Matthieu Crussière et Luc Le Magoarou, "CSI Compression Using Channel Charting", *à 2024 58th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, USA, 2024, pp. 1660-1664

(C3) Baptiste Chatelier, Luc Le Magoarou, Vincent Corlay et Matthieu Crussière, "Model-Based Learning for Location-to-Channel Mapping", *à 2024 IEEE International Conference on Acoustics, Speech and Signal Processing*, Seoul, Republic of Korea, 2024, pp. 12836-12840

(C4) Taha Yassine, Baptiste Chatelier, Vincent Corlay, Matthieu Crussière, Stéphane Paquelet, Olav Tirkkonen et Luc Le Magoarou, "Model-Based Deep Learning for Beam Prediction Based on a Channel Chart", *à 2023 57th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, USA, 2023, pp. 1636-1640

(C5) Baptiste Chatelier, Luc Le Magoarou et Getachew Redieteab, "Efficient Deep Unfolding for SISO-OFDM Channel Estimation", *à IEEE International Conference on Communications*, Rome, Italy, 2023, pp. 3450-3455

Les publications suivantes de l'auteur, réalisées au cours de la thèse, à l'exception de (C9), ne sont pas incluses dans le manuscrit.

Articles de conférence

(C6) José Miguel Mateos-Ramos, Baptiste Chatelier, Christian Häger, Musa Furkan Keskin, Luc Le Magoarou et Henk Wymeersch, "Semi-Supervised End-to-End Learning for Integrated Sensing and Communications," *à 2024 IEEE International Conference on Machine Learning for Communication and Networking*, Stockholm, Sweden, 2024, pp. 132-138

(C7) Taha Yassine, Luc Le Magoarou, Baptiste Chatelier, Stéphane Paquelet et Matthieu Crussière, "Cartographie du canal par réduction de dimension et réseaux triplets", *à 29° Colloque sur le traitement du signal et des images, GRETSI*, Grenoble, France, 2023, pp. 1165-1168

(C8) Baptiste Chatelier, Vincent Corlay, Cristina Ciochina, Falou Coly et Julien Guillet, "Influence of Dataset Parameters on the Performance of Direct UE Positioning via Deep Learning," *à 2023 Joint European Conference on Networks and Communications & 6G Summit*, Gothenburg, Sweden, 2023, pp. 126-131

(C9) Baptiste Chatelier et Matthieu Crussière, "On the Impact of Phase Noise on Beamforming Performance for mmWave Massive MIMO Systems," *à 2022 IEEE Wireless Communications and Networking Conference*, Austin, USA, 2022, pp. 1563-1568

Mobilité internationale

Dans le cadre de cette thèse, une mobilité internationale a été réalisée entre Septembre et Décembre 2025, dans le groupe *Système de communications* de l'université de technologie de Chalmers à Göteborg en Suède. Plus particulièrement, l'équipe d'accueil était l'équipe CROSSNET du Prof. Henk Wymeersch. Le travail présenté dans [Cha+25c] a été préparé pendant cette période, en collaboration avec le Prof. Henk Wymeersch, l'Assist. Prof. Chritian Häger et José Miguel Mateos-Ramos.

Introduction

Context

Since the appearance of the earliest forms of the human species, humans have developed increasingly complex methods to communicate between each others. This process began with the acquisition of speech by *Homo sapiens*, which laid the foundation for the emergence of structured languages. Visual modes of communication followed, initially through forms of parietal art, and later through the development of early writing systems such as cuneiform and hieroglyphs. These systems enabled humans to record, store, and transmit information across time and space. The introduction of alphabets further led to the stabilization of languages within regions, contributing to increased literacy, access to education, and increased efficiency in all forms of social activities, from agriculture to art and medicine. A major turning point came with the invention of the printing press in the 15th century, a groundbreaking innovation that enabled mass production of books, and significantly expanded the dissemination of knowledge. However, books as a communication medium exhibit several limitations: they require literacy, are not universally accessible, and allow for relatively slow information dissemination and updates.

Rapid transmission of information over long distances began with basic systems such as smoke signals and light beacons, as early as the 7th century BCE. Modern long distance transmission systems can be traced back to Claude Chappe's optical telegraph, developed in 1790. This basic mechanical system significantly reduced information transmission times: for instance, transmitting a message between Paris and Strasbourg, two cities in France distant of approximately 500km, went from four days by horse to merely two hours with Chappe's telegraph. From then, subsequent rapid scientific and technological progress led to the development of electromagnetic telegraphy, with foundational contributions of Samuel Thomas von Sömmerring (1809), Francis Ronalds (1816), André-Marie Ampère (1820), William Fothergill Cook and Charles Wheatstone (1837), and Samuel Morse (1838). In 1876, Alexander Graham Bell invented the telephone, enabling the first transcontinental phone call by the end of 1914. Both the telegraph and the telephone were *wired communication* mediums as they relied on electric wires to transmit information. Consequently, these technologies faced inherent limitations, including high deployment and maintenance costs, as well as vulnerability to transmission medium failures.

In the 1890s, Guglielmo Marconi pioneered *wireless communications* by combining radio wave technology, theorized by James Clerk Maxwell (1865) and confirmed experimentally by Heinrich Hertz (1887), with telegraphic principles, leading to the development of the first wireless telegraphy system. This innovation eventually evolved into what is known today as radio systems. The two World Wars and the later Cold War in the 20th century spurred a frenetic race for technological innovation. This period saw major

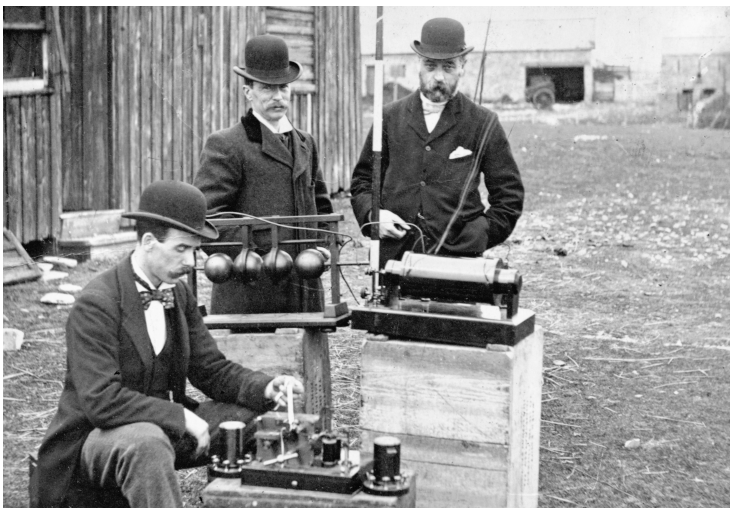


Figure 1: British engineers in front of Marconi's radio equipments during a presentation in 1897 (courtesy of [Wik25b]).

advances in the wireless communications field, with key innovations such as radio detection and ranging (RADAR) with major contributions of Robert Watson Watt and Hans Hollmann in the 1930s, or the development of satellite communications and the global positioning system (GPS) in the 1970s. The 1980s marked the beginning of the era of wireless cellular networks generations. The first generation (1G), around 1980, supported only analog audio. The second generation (2G), around 1990, introduced the support of digital signals and text messages. The third generation (3G), around 2000, increased data transmission rates and introduced the support of mobile internet, allowing users to connect to the internet via mobile devices. The fourth generation (4G), around 2010, further increased data rates, enabling real-time wireless video transmission. The fifth generation (5G), around 2020, brought significantly higher data rates, increased bandwidth, and high-reliability low-latency communication capabilities. The sixth generation (6G), expected around 2030, aims to further advance data transmission capabilities. In addition to continued improvements in speed and bandwidth, 6G is also expected to integrate sustainability considerations into its design, and to adopt artificial intelligence/machine learning (AI/ML) as a core component of its future functionalities.

The past decade has witnessed a significant surge in the application of machine learning techniques across a broad spectrum of problems. This trend is illustrated by the 2024 Nobel Prize in Physics, awarded to John Hopfield and Geoffrey Hinton in recognition of their pioneering contributions to artificial neural networks. Fields such as the automotive industry, healthcare, finance, or telecommunications, have increasingly integrated machine learning methods in their standard practices. This adoption is not limited to professional settings: the general public has recently adopted AI methods through conversational agents such as ChatGPT, LeChat, or Claude, or image generation tools like DALL-E or Midjourney. This recent interest increase can be attributed to several factors.

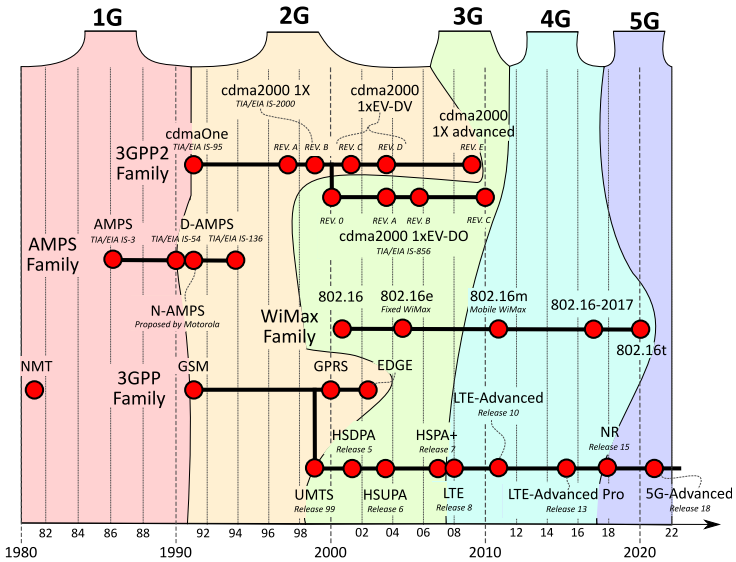


Figure 2: Illustration of the continuous evolution of cellular networks standards over the past 40 years (courtesy of [Wik25c]).

First, the technological developments in the microelectronics field over the past thirty years have enabled the access to efficient and powerful computing resources, which has enabled the developments of increasingly complex AI/ML methods. Indeed, while the ML field can be traced back to the 1960s, progress at that time was limited by the lack of sufficiently powerful computational resources. Second, by definition ML methods require a training phase, which may require large scale datasets. While humanity has always generated knowledge and information, the development and rapid expansion of internet over the past three decades have significantly increased the amount of available data. For instance, it is estimated that, in 2025, around 400 billion gigabytes of data is generated every day [Des19]. Together, these developments have significantly enhanced the performance, usability, and impact of machine learning systems in both specialized and everyday contexts.

Despite this rapid growth, and its increasing use to solve a wide range of problems, ML methods still face several key limitations. First, modern models are characterized by substantial computational complexities. For instance, large language models (LLMs) used in conversational agents can contain up to trillions of parameters, all of which must be optimized during a training phase. This results in extremely long training times and associated costs: as an illustrative example, modern LLMs require several months of training on complex distributed computational architectures, with associated costs reaching several million USD. Second, current models require extremely large scale datasets to achieve their high levels of performance. Although the volume of available data has grown exponentially in recent years, some researchers anticipate that the entirety of publicly accessible human-generated textual data may be exhausted for LLM training before 2030. This projection raises concerns regarding the scalability and sustainability of future model

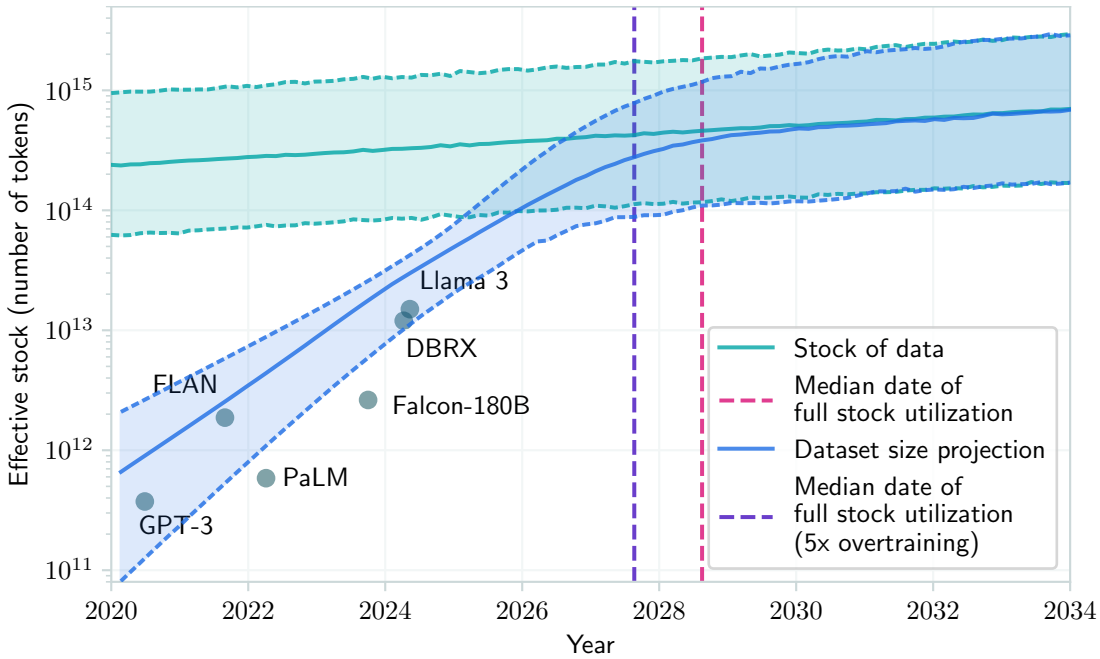


Figure 3: Illustration of the data availability issue for deep learning models training: large language models are expected to have been trained on the totality of public human-generated textual data before 2030 (Fig. 1 in [Vil+24]).

developments. Third, due to their extreme complexity, modern ML models suffer from limited interpretability. As a result, they are often referred to as *black boxes*: in many cases, it is fundamentally unclear why a given input leads to a particular output.

In recent years, significant attention has been devoted to enhancing the interpretability of machine learning models, aiming to mitigate their black-box nature. One of the key research directions in this area is the *model-based machine learning paradigm*. This paradigm seeks to combine the rigor and structure of mathematical models traditionally used in signal processing, with the flexibility and learning capabilities of modern machine learning methods. Indeed, in the signal processing field, proposed solutions typically rely on simple mathematical models, carefully crafted based on prior knowledge of the studied problem. This prior knowledge usually originates from an understanding of the physical principles governing the studied system, thereby providing an interpretable basis for the resulting solution. Owing to their dependence on a given mathematical model, these approaches are referred to as *model-based*. However, as observed by George E. P. Box, *"All models are wrong, but some are useful"*: the mathematical models used in signal processing are generally built upon simplifying assumptions, inherently limiting their validity. As a result, even minor deviations from the originally studied system may lead to solutions, derived from simplified models, that are suboptimal. In contrast, machine learning methods are referred to as *data-driven*, as they learn solutions directly from data. This confers increased flexibility and robustness to minor deviations from the assumed system

model. However, as previously mentioned, this flexibility generally comes at the expense of interpretability. The model-based machine learning paradigm thus aims to combine the strengths of both approaches: leveraging the adaptability of data-driven methods while retaining the interpretability and structure provided by model-based methods.

Wireless telecommunications require signal processing methods, which rely on mathematical models to describe and optimize wireless information transmission. While ML methods have recently been applied to enhance certain communication techniques, they also raise significant interpretability and complexity concerns. Given both the abundance of established mathematical models, and the limitations of purely data-driven approaches, the wireless communications field appears as particularly well adapted to the use of the model-based machine learning paradigm. This observation motivates the central question of this thesis, as presented below.

Central research question

Can the model-based machine learning paradigm be used to provide interpretable and low-complexity solutions to challenging wireless communication problems?

Thesis outline

The aforementioned question is inherently broad, and addressing it comprehensively would require evaluating model-based machine learning methods across the full spectrum of wireless communication problems, an undertaking that lies beyond the feasible scope of a single thesis. Therefore, this thesis focuses on three main research axes, which are presented below.

Research axes

(A1) **Location-to-channel mapping.** Is it possible to design a neural architecture that learns the mapping between a location and the associated propagation channel? If so, is it possible to use this neural architecture to enhance the precision of wireless localization methods?

(A2) **Hardware impairments compensation.** In actual wireless communication systems, hardware impairments lead to suboptimal transmission performance: is it possible to use model-based machine learning methods to compensate their impact?

(A3) **Channel compression.** In modern wireless cellular networks, the high dimensionality of propagation channel coefficients entails significant overhead. Can channel charting, a specific model-based dimensionality reduction method, be used to reduce the complexity?

The objective of this thesis is to advance the current state of knowledge across the three identified research axes. This manuscript is structured into two parts, and organized in nine chapters, as outlined below. Each chapter is designed to be self-contained, allowing it to be read independently of the others. However, the author strongly recommends that the reader consult Chapter 3, which provides a detailed introduction to the core concepts of this thesis.

- Part I: Background and preliminary concepts
  - Chapter 1 introduces fundamental concepts related to the propagation channel. Specifically, it is shown how a mathematical model of the propagation channel can be derived based on the physical principles describing wave-matter interactions. Subsequently, this chapter presents several examples illustrating how modern communication techniques leverage propagation channel knowledge to enhance transmission performance.
  - Chapter 2 provides a general introduction to the machine learning field. After presenting the different learning problem types and fundamental learning notions, this chapter presents in details three popular machine learning methods: linear regression, k-nearest neighbors and principal component analysis. A particular emphasis is then placed on deep learning, as it constitutes the core of most modern machine learning methods. As such, general theoretical notions regarding neural networks and associated training procedures are introduced. Finally, this chapter discusses about the current known limitations of deep neural networks.
  - Chapter 3 presents the model-based machine learning paradigm in details. In particular, it highlights how this approach combines the strengths of model-based and machine learning approaches, effectively enhancing model interpretability and coping with domain knowledge deficiency. The chapter then presents how this paradigm can be applied in the problems described in research axes (A1)-(A3), and thus serves as an introduction to the contributions of this thesis.
- Part II: Contributions
  - Chapter 4 focuses on (A1): the design of a model-based neural network that learns the location-to-channel mapping is presented. This contribution builds on the concept of implicit neural representations, where neural networks are used to approximate continuous functions. The proposed neural network is directly obtained from derivations on a physics-based propagation channel model, thereby enabling an interpretable structure and explainable learning performance. Experimental results demonstrate that the proposed model successfully learns the desired mapping and, in contrast to classical neural baselines,

overcomes the spectral bias issue: it is able to capture both the low and high-frequency components of the mapping, which can be explained based on the structure of the proposed neural model.

- Chapter 5 addresses (A1) by applying the neural network proposed in Chapter 4 to the wireless localization problem. In particular, the network is used to enhance the classical fingerprinting localization method by serving as a generative neural channel model. By leveraging its ability to infer channel coefficients at arbitrary locations, the network augments the fingerprinting database, thereby improving localization accuracy. Furthermore, theoretical developments on the similarity measure used to perform fingerprinting are exploited to propose a model-based enhanced localization method, further improving localization accuracy. Extensive experiments demonstrate that the proposed method achieves sub-wavelength accuracy, even in challenging non-line-of-sight scenarios.
- Chapter 6 focuses on (A2) by improving a model-based neural network designed for channel estimation and hardware impairments learning. The existing network is used in a multi-frequency wireless communication system, to learn antenna gain impairments and frequency offsets. Both its number of learnable parameters and training time are optimized following the model-based machine learning paradigm. Experiments showcase the ability of the proposed method to achieve efficient hardware impairments learning and precise channel estimation.
- Chapter 7 develops (A2) by introducing a differentiable algorithm that jointly performs direction of arrival (DoA) estimation and antenna hardware impairments learning. The proposed model-based approach builds on the well-know MUSIC algorithm, extending it into a differentiable framework. Furthermore, task-adapted supervised and unsupervised loss functions are proposed to enhance learning performance, and experiments illustrate the superior performance of the proposed method against the non-differentiable MUSIC algorithm.
- Chapter 8 is dedicated to (A3), with the aim of using channel charting, an unsupervised dimensionality reduction method, to reduce the beam management complexity in cell-free massive MIMO systems. By leveraging channel charting, the channel state information is compressed into a low-dimensional pseudo-location representation, which is subsequently shared across the different base stations. At each base station, a task-specific neural network is used to decode the pseudo-location into the optimal beam index within a predefined codebook, thereby enabling efficient beamforming. Experimental evaluations demonstrate the effectiveness of the proposed approach, highlighting its performance advantages over different baselines.
- Chapter 9 covers (A3) with the objective of using channel charting to perform task-based channel compression within massive MIMO systems. This contri-

bution focuses on encoding channel state information at the UE level through channel charting, transmitting the resulting low dimensional pseudo-locations to the base station, and subsequently decoding them into a precoding matrix with a well-adapted neural network. A heuristic learning strategy is proposed for this task-based channel state information compression framework, and its performance is explored with different compression ratios, demonstrating superior performance compared to baselines.

In Part II, the computational cost for each contribution is provided in GPU hours at the beginning of each chapter. The computations were primarily conducted on a cluster of NVIDIA A40 GPUs, thereby, the reported times represent the equivalent computational time on a single NVIDIA A40 GPU.

List of publications

The work presented in this manuscript is based on the following publications.

Journal articles

(J1) Baptiste Chatelier, Vincent Corlay, Musa Furkan Keskin, Matthieu Crussière, Henk Wymeersch and Luc Le Magoarou, "Model-based Implicit Neural Representation for sub-wavelength Radio Localization", *submitted to the IEEE Transactions of Wireless Communications*, June 2025; currently under review

(J2) Baptiste Chatelier, Vincent Corlay, Matthieu Crussière and Luc Le Magoarou, "Model-Based Learning for Multi-Antenna Multi-Frequency Location-to-Channel Mapping", *in IEEE Journal of Selected Topics in Signal Processing*, vol. 19, no. 3, pp. 520-535, April 2025

Conference articles

(C1) Baptiste Chatelier, José Miguel Mateos-Ramos, Vincent Corlay, Christian Häger, Matthieu Crussière, Henk Wymeersch and Luc Le Magoarou, "Physically Parameterized Differentiable MUSIC for DoA Estimation with Uncalibrated Arrays", *in IEEE International Conference on Communications*, Montréal, Canada, 2025

(C2) Baptiste Chatelier, Vincent Corlay, Matthieu Crussière and Luc Le Magoarou, "CSI Compression Using Channel Charting", *in 2024 58th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, USA, 2024, pp. 1660-1664

(C3) Baptiste Chatelier, Luc Le Magoarou, Vincent Corlay and Matthieu Crussière, "Model-Based Learning for Location-to-Channel Mapping", *in 2024 IEEE International Conference on Acoustics, Speech and Signal Processing*, Seoul, Republic of Korea, 2024, pp. 12836-12840

(C4) Taha Yassine, Baptiste Chatelier, Vincent Corlay, Matthieu Crussière, Stéphane Paquelet, Olav Tirkkonen and Luc Le Magoarou, "Model-Based Deep Learning for Beam Prediction Based on a Channel Chart", *in 2023 57th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, USA, 2023, pp. 1636-1640

(C5) Baptiste Chatelier, Luc Le Magoarou and Getachew Redieteab, "Efficient Deep Unfolding for SISO-OFDM Channel Estimation", *in IEEE International Conference on Communications*, Rome, Italy, 2023, pp. 3450-3455

The following publications by the author, produced during the course of this thesis, with the exception of (C9), are not included in this manuscript.

Conference articles

(C6) José Miguel Mateos-Ramos, Baptiste Chatelier, Christian Häger, Musa Furkan Keskin, Luc Le Magoarou and Henk Wymeersch, "Semi-Supervised End-to-End Learning for Integrated Sensing and Communications," *in 2024 IEEE International Conference on Machine Learning for Communication and Networking*, Stockholm, Sweden, 2024, pp. 132-138

(C7) Taha Yassine, Luc Le Magoarou, Baptiste Chatelier, Stéphane Paquelet et Matthieu Crussière, "Cartographie du canal par réduction de dimension et réseaux triplets", *in 29° Colloque sur le traitement du signal et des images, GRETSI*, Grenoble, France, 2023, pp. 1165-1168

(C8) Baptiste Chatelier, Vincent Corlay, Cristina Ciochina, Falou Coly and Julien Guillet, "Influence of Dataset Parameters on the Performance of Direct UE Positioning via Deep Learning," *in 2023 Joint European Conference on Networks and Communications & 6G Summit*, Gothenburg, Sweden, 2023, pp. 126-131

(C9) Baptiste Chatelier and Matthieu Crussière, "On the Impact of Phase Noise on Beamforming Performance for mmWave Massive MIMO Systems," *in 2022 IEEE Wireless Communications and Networking Conference*, Austin, USA, 2022, pp. 1563-1568

International mobility

In the context of this thesis, an international mobility was organized between September and December 2024, in the *Communication systems* group at Chalmers University of Techology in Göteborg, Sweden. Specifically, this mobility took place in the CROSSNET team led by Prof. Henk Wymeersch. The work presented in [Cha+25c] was prepared during that period, in collaboration with Prof. Henk Wymeersch, Assist. Prof. Christian Häger, and José Miguel Mateos-Ramos.

Part I

Background and preliminary concepts

Chapter 1

Propagation channel

Over the past five decades, a deeper understanding of wireless propagation phenomena has led to significant advancements in the field, enhancing both the efficiency and performance of modern wireless communication systems. Consequently, a thorough grasp of the physical mechanisms governing the propagation channel is fundamental to the study of contemporary signal processing techniques used in wireless communications. Section 1.1 describes physical phenomena that govern the propagation channel. Section 1.2 builds on those phenomena and presents usual channel models. Finally, Section 1.3 shows how the propagation channel knowledge can be used in modern wireless communication systems, and Section 1.4 presents a brief conclusion.

Contents

1.1 Propagation channel and physical phenomena **14**

1.1.1 Notion of propagation channel 14

1.1.2 Physical phenomena 14

1.2 Mathematical model of the propagation channel **20**

1.2.1 A ray-based approach 20

1.2.2 Impact of physical phenomena on the propagation channel . . . 23

1.3 Application to modern communication systems **31**

1.3.1 Channel capacity 31

1.3.2 Multi-antenna systems 32

1.4 Conclusion **45**

This chapter is inspired by the very good book of Tse [TV05], as well as the one of Björnson and Demir [BD24], which serve as excellent references for readers seeking to gain knowledge in the principles of wireless communications and in multiple antenna systems.

1.1 Propagation channel and physical phenomena

1.1.1 Notion of propagation channel

In 1948, Claude Shannon published his well-known paper *"A mathematical theory of communications"* [Sha48], introducing one of the foundational models of communications. In this model, illustrated in Fig. 1.1, the propagation channel represents the medium through which a message travels from a transmitter to a receiver, encompassing all intermediate elements, and being affected by noise. In the context of wireless communications, the notion of propagation channel can be further specified: it corresponds to the physical environment through which electromagnetic waves, emitted by a transmitting antenna system, propagate towards a receiving antenna system. Within this physical environment, the transmitted electromagnetic waves are subject to several phenomena such as attenuation, reflection, refraction, diffraction, scattering, and Doppler shift, all of which are explicitly defined in the next subsection.

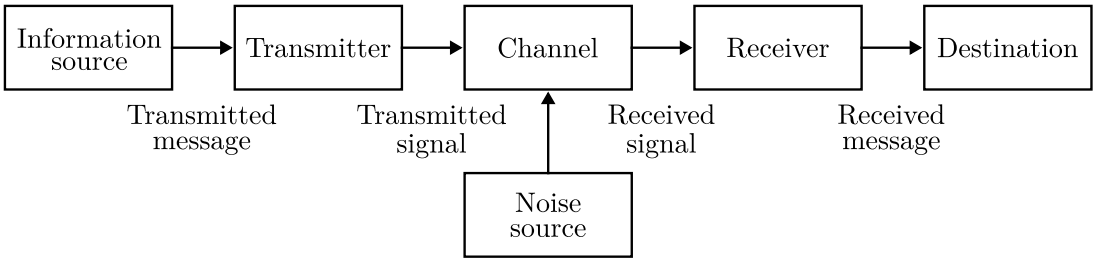


Figure 1.1: Shannon's model.

1.1.2 Physical phenomena

Depending on the transmitter/receiver antenna spatial configuration, some phenomena may or may not arise [1]. The simplest scenario happens when the electromagnetic waves propagate in the *free space* regime.

Free space propagation

A formal definition of the free-space propagation regime is presented in Highlight 1.1. This regime requires the region between the transmitter and the receiver to be entirely free of obstacles within a specific clearance zone, referred to as the first Fresnel zone. When this condition is satisfied, propagation occurs under free-space conditions, commonly referred to as line-of-sight (LoS). In this case, an emitted wave is subject only to noise and attenuation, which denotes the reduction in its amplitude as a function of the propagation distance, typically following an inverse proportionality law in free-space conditions. In contrast, in more complex environments, additional phenomena may occur, depending on

1. See [BD24, Chapter 5, pp. 309-312] for a detailed explanation

the nature and distribution of physical objects interacting with the propagating wave. This is illustrated in Fig. 1.3, and explained in the next paragraphs.

Remark 1.1. *Given a transmitter/receiver distance of* $10m$*, with an operating frequency of* $2.4GHz$ *(classical WiFi frequency), the required clearance for free space propagation is* $r_{\mathsf{cl}} \simeq 0.5m$*, which can be obtained in certain indoor environments. However, taking a distance of* $10km$ *with an operating frequency of* $3.5GHz$ *(classical wireless cellular network frequency), the required clearance becomes* $r_{\mathsf{cl}} \simeq 14.6m$*, which becomes challenging. For long transmitter-receiver distances, the Earth's curvature must also be taken into account.*

Highlight 1.1: Free space propagation regime

Propagation happens in the free space regime if the first Fresnel zone between the transmit and receive antennas is clear from any physical obstacle. This zone, illustrated in Fig. 1.2, depends on the distance between the transmitter and receiver D, on the transmission frequency f, with wavelength $\lambda = c/f$. It is properly defined as the ellipsoid given by the set:

$$\left\{\mathbf{x} \in \mathbb{R}^3 | d_1(\mathbf{x}) + d_2(\mathbf{x}) - D \leq \frac{\lambda}{2}\right\}, \tag{1.1}$$

where $d_1(\mathbf{x})$ is the distance between the transmitter antenna and $\mathbf{x}$, $d_2(\mathbf{x})$ is the distance between $\mathbf{x}$ and the receiver antenna, and D is the transmitter-receiver distance. The equatorial radius of this ellipsoid, i.e. maximum of the semi-minor axis length, also known as the required clearance and denoted as r_{cl}, is obtained for $d_1(\mathbf{x}) = d_2(\mathbf{x})$, which yields:

$$r_{\mathsf{cl}} \simeq \frac{1}{2}\sqrt{\lambda D}. \tag{1.2}$$

A proof of Eq. (1.2) is provided in Appendix A.1.

Reflection - Absorption - Refraction

When a propagating wave interacts with an object whose dimensions are large and whose surface is relatively smooth compared to the operating wavelength, phenomena such as specular reflection, refraction, and absorption occur. In such cases, a portion of the incident wave is absorbed by the object, while the remainder is divided into a reflected and refracted component, in accordance with Snell-Descartes laws, which are presented in Highlight 1.2 and illustrated in Fig. 1.4. Both the reflected and refracted waves are typically attenuated and may exhibit a phase shift relative to the incident wave. In wireless cellular networks, such phenomena commonly occur in urban environments, where

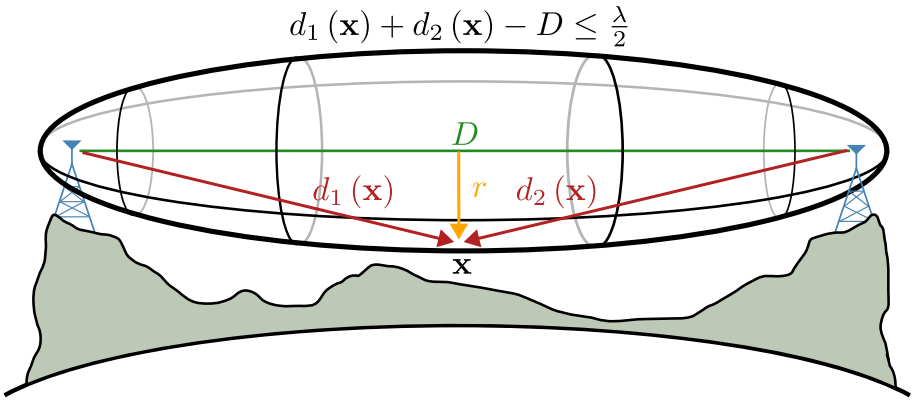


Figure 1.2: Illustration of the first Fresnel zone (based on [Wik25a]).

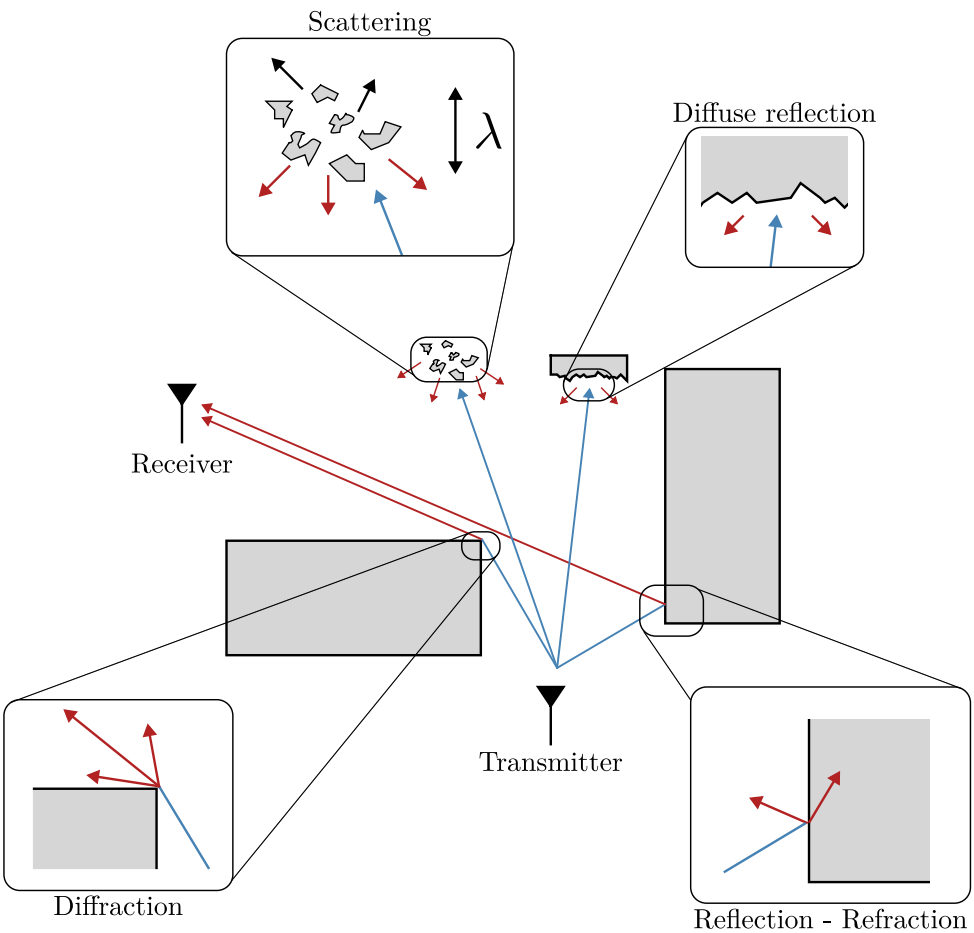


Figure 1.3: Illustration of the different physical phenomena when considering obstacles between the transmitter and receiver: here the receiver is in the non line-of-sight zone of the emitter (based on Fig. 3.2 in [Col03]).

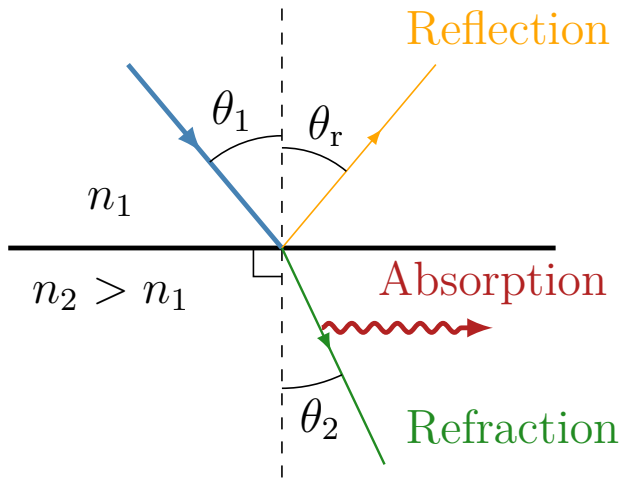


Figure 1.4: Illustration of the reflection/absorption/refraction phenomena (based on [Neu25c]).

buildings can reflect a portion of the incident electromagnetic waves, while also absorbing and refracting significant energy depending on the material properties and the operating frequency.

Highlight 1.2: Snell-Descartes laws

Given an incident wave of angle θ_1, impinging on the interface between a medium of refractive index n_1, and another medium of refractive index n_2, the Snell-Descartes laws of reflection and refraction can be defined as:

$$\begin{cases} \text{(Reflection)} & \theta_1 = \theta_r \\ \text{(Refraction)} & n_1 \sin(\theta_1) = n_2 \sin(\theta_2) \end{cases} \tag{1.3}$$

Both reflection and refraction phenomena depend on the angle of incidence of the incoming wave. When the incident wave impinges normally on a perfect conductor, it is entirely reflected, with no transmission into the medium. In contrast, refraction occurs when a wave passes from one medium into another with a different refractive index. The degree of bending and the fraction of transmitted energy depend on the difference between the refractive indices of the two media.

Diffraction

Diffraction occurs under specific conditions when an incident electromagnetic wave encounters an object that neither significantly absorbs nor reflects the wave energy. These

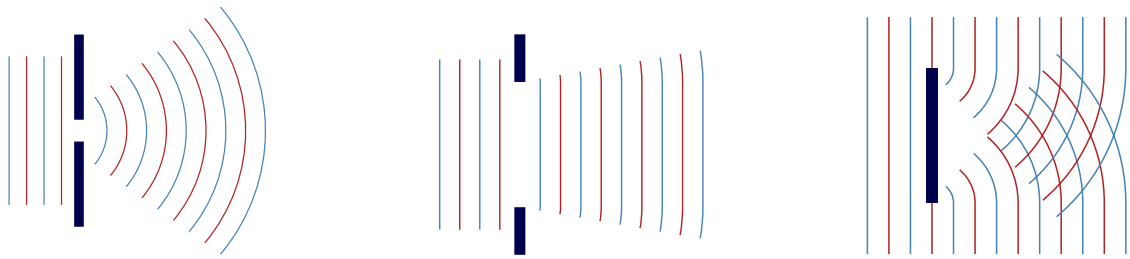


Figure 1.5: Illustration of the diffraction phenomenon: small/large slits with respect to the wavelength, and edges diffraction (based on [Neu25a]).

conditions are presented in details in Highlight 1.3, and illustrated in Fig. 1.5. Under these conditions, the wave bends around the obstacle or spreads after passing through an aperture, giving rise to a diffracted wave.

Highlight 1.3: Diffraction conditions

For diffraction to occur, at least one of the following conditions must be met:

1. The obstacle is small relative to the operating wavelength.
2. The obstacle contain slits or apertures that are narrow compared to the wavelength.
3. The obstacle presents sharp edges.

Under these conditions, the wavefront is disturbed in a manner that causes it to bend or spread, resulting in the generation of a diffracted wave.

In the context of wireless communications, a common example of diffraction is the bending of electromagnetic waves around the edges of buildings or other urban structures, enabling signal propagation into areas that are in non line-of-sight (NLoS) with respect to the transmitter.

Scattering

Scattering occurs when an incident wave interacts with multiple objects whose dimensions are comparable to, or smaller than, the operating wavelength. These scattering elements may include physical obstacles, such as leaves in trees, or inhomogeneities in the refractive index between different media. As a result of these interactions, the incident wave is redistributed in multiple directions: see Fig. 1.6 for an illustration. Depending on the physical properties, shape, and spatial distribution of the scattering objects, preferred scattering directions may emerge, resulting in anisotropic scattering patterns, as observed

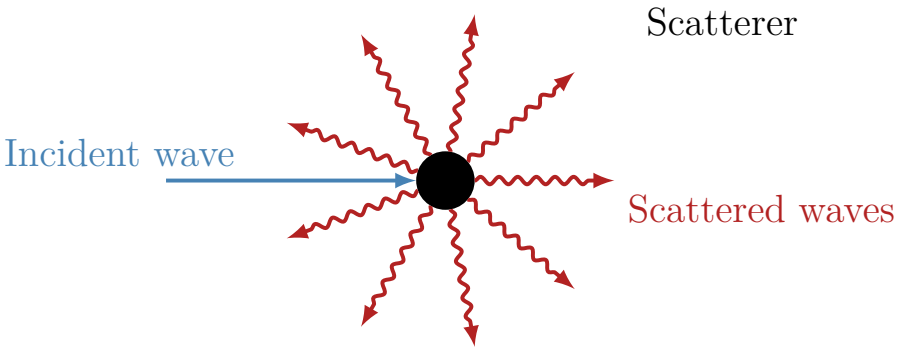


Figure 1.6: Illustration of the scattering phenomenon.

in the Mie scattering phenomenon. The scattering phenomenon is responsible for many everyday observations, such as the blue color of the sky and the white appearance of clouds, both resulting from specific scattering processes. When an incident wave interacts with an object whose surface is large but rough relative to the operating wavelength, diffuse reflection occurs. In this case, the reflected energy is distributed over multiple directions rather than being concentrated along a single specular path. This behavior can be considered a particular form of scattering.

Doppler shift

When considering a moving transmitter, or receiver, the observed signal frequency at the receiver will differ from the emitted one: this is a manifestation of the Doppler effect, which is presented more formally in Highlight 1.4. This phenomenon can be illustrated for audio waves by the well-known ambulance example: as an ambulance approaches a stationary observer, the observer perceives an increased frequency in the horn sound (corresponding to a decreasing transmitter-receiver distance, $f > f_0$) [2]. Conversely, as the ambulance moves away after passing the observer, the observed frequency decreases (corresponding to an increasing transmitter-receiver distance, $f < f_0$). It is of particular importance in wireless communication, as Doppler shifts arise in many examples:

- **Wireless connectivity in high speed trains.** The very high velocity of modern trains results in a significant relative speed between a train passenger and a fixed roadside transmitter, giving rise to particularly large Doppler shifts.
- **Wireless cellular networks.** In urban environments, moving obstacles such as vehicles introduce time variations in the channel and induce Doppler shifts along NLoS propagation paths.

2. f_0 being the horn sound frequency, and f the perceived sound frequency.

Highlight 1.4: Doppler effect

Considering a medium in which the wave propagation speed is c, with v_{tr} and v_{re} respectively denoting the speeds of the transmitter and receiver, the observed signal frequency f at the receiver is given by:

$$f = \left(1 + \frac{\Delta v}{c}\right) f_0, \tag{1.4}$$

where f_0 denotes the emitted signal frequency, and $\Delta v = -(v_{\mathrm{re}} - v_{\mathrm{tr}})$ denotes the opposite of the relative speed of receiver and transmitter. Two cases can be accounted for:

1. The transmitter-receiver distance is decreasing: $\Delta v > 0 \Rightarrow f > f_0$
2. The transmitter-receiver distance is increasing: $\Delta v < 0 \Rightarrow f < f_0$

1.2 Mathematical model of the propagation channel

1.2.1 A ray-based approach

Given a transmitter/receiver spatial configuration, it was shown in Section 1.1.1 that if the first Fresnel zone is obstructed by physical obstacles, the emitted wave will propagate partially or entirely in a NLoS regime. Depending on the nature of the physical obstacles present in the environment, the emitted wave undergoes various propagation phenomena, as presented in Fig. 1.3. This results in multiple versions of the emitted signal arriving at the receiver with different delays and attenuations, a phenomenon known as *multipath propagation*. A key characteristic of the wireless channel is its number of distinguishable propagation paths [3], with each path representing a distinct link between the transmitter and receiver. Note that a link may be either direct, i.e. LoS propagation, or indirect, i.e. NLoS propagation. Given the local and complex nature of the phenomena operating in the NLoS regime [4], indirect propagation paths can be decomposed as a set of *rays*, in a manner analogous to optical geometry. For an indirect propagation path, the total delay and attenuation coefficients arise from the cumulative contributions of the constituent rays. This ray-based channel model is illustrated in Fig. 1.7. Complete knowledge of the channel can, in principle, be obtained by fully characterizing the interactions of each

3. Two propagation paths are considered distinct if they are resolvable by a measurable difference in one or more of their fundamental signal properties, e.g. distinct Doppler shifts, or distinct propagation delays.

4. For instance, in the case of scattering, the incident wave is decomposed into multiple waves that may subsequently interact with other physical elements.

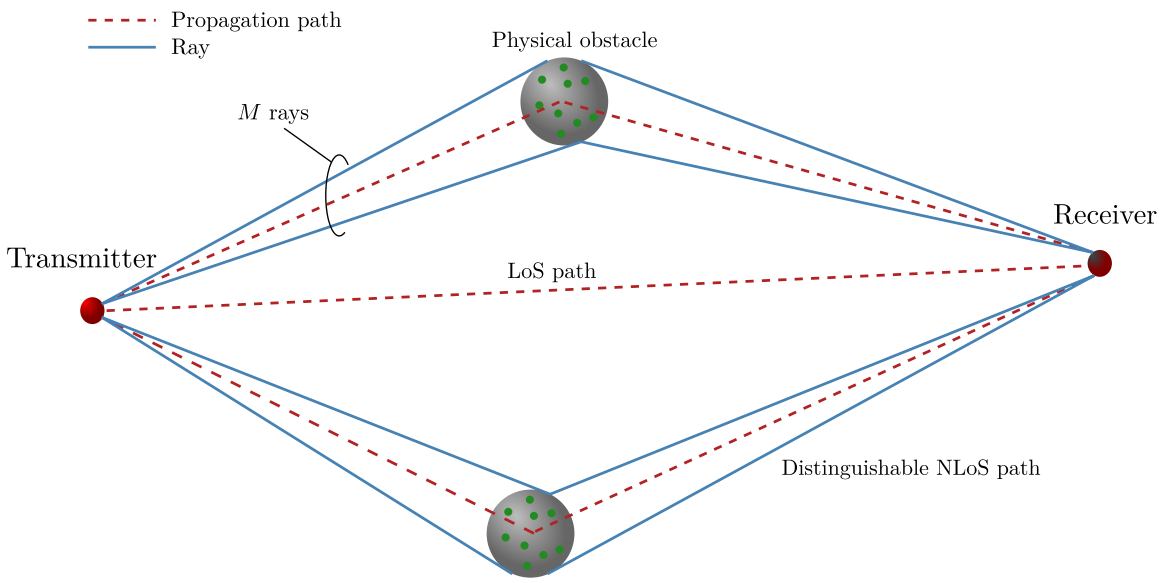


Figure 1.7: Ray-based channel model (based on Fig. 3.1 in [Fon22]).

ray with the propagation environment: this constitutes the core idea behind *ray-tracing* channel simulation techniques [Hoy+22].

As an example, one can consider a simple propagation environment with a LoS path and one NLoS path originating from a reflecting obstacle. The attenuation coefficients are denoted by α_{LoS} and α_{NLoS}. Given an emitted wave $x(t)$, the received signal $y(t)$ can be expressed as:

$$y(t) = \alpha_{\text{LoS}} x(t - \tau_{\text{LoS}}) + \alpha_{\text{NLoS}} x(t - \tau_{\text{NLoS}}). \tag{1.5}$$

with $\tau_{\text{LoS}} = d_{\text{LoS}}/c$, where d_{LoS} denotes the LoS propagation path distance. Note that the delay τ_{NLoS} arises from both the cumulative effect of the NLoS propagation path distance and a phase-shift induced by wave-matter interactions on this path. From Eq. (1.5), it becomes evident that the channel can be described in terms of the attenuation coefficients and delays of its distinguishable propagation paths. Formally, the channel can be modeled as a linear time-varying system, whose impulse response is presented in Definition 1.1.

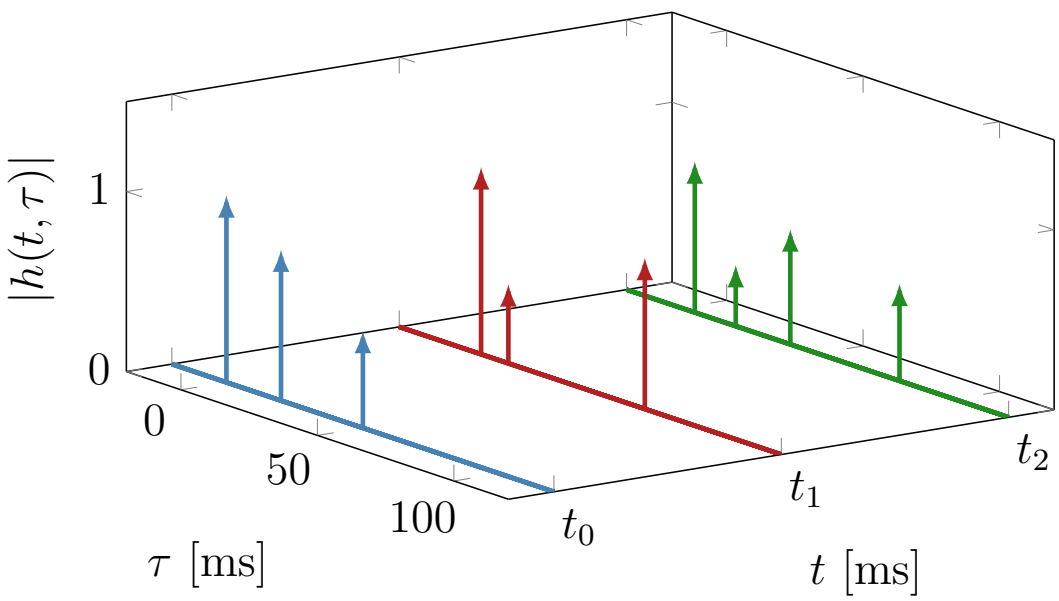


Figure 1.8: Time evolution of the channel impulse response.

Definition 1.1: Channel impulse response

Let L_p be the number of distinguishable propagation paths of a wireless channel. Its time-varying impulse response is defined as:

$$h(t,\tau) = \sum_{l=1}^{L_p} \alpha_l(t)\,\delta(\tau - \tau_l(t))\,\mathrm{e}^{-\mathrm{j}2\pi f_{d,l} t}, \tag{1.6}$$

where $\alpha_l(t) \in \mathbb{C}$, $\tau_l(t) \in \mathbb{R}^+$, and $f_{d,l} \in \mathbb{R}$ respectively denote the complex attenuation coefficient, propagation delay and Doppler frequency of the lth distinguishable path.

The time-varying nature of the channel arises from changes in the set of distinguishable propagation paths over time. Due to potential mobility within the propagation environment, e.g. moving vehicles or pedestrians, certain propagation paths that exist at time t may no longer be present at time $t + \Delta t$. This dynamic behavior affects both the number of distinguishable paths, and their associated delays and attenuation coefficients, leading to a continuously evolving channel impulse response (CIR), which is illustrated in Fig. 1.8. Characterizing a propagation environment can also be achieved through the analysis of its frequency response. For a given transmitted signal, the channel frequency response provides insight into the amplitude and phase distortion introduced by the channel across the frequency domain. It is obtained by taking the Fourier transform of the channel

impulse response along the delay dimension, and is presented in Definition 1.2, with a proof proposed in Appendix A.2.

Definition 1.2: Channel frequency response

Let L_p be the number of distinguishable propagation paths of a wireless channel. Its frequency response is defined as:

$$h(t, f) = \sum_{l=1}^{L_p} \alpha_l(t) \, \mathrm{e}^{-\mathrm{j}2\pi f \tau_l(t)} \mathrm{e}^{-\mathrm{j}2\pi f_{d,l} t}, \tag{1.7}$$

where $\alpha_l(t) \in \mathbb{C}$, $\tau_l(t) \in \mathbb{R}^+$, and $f_{d,l} \in \mathbb{R}$ respectively denote the complex attenuation coefficient, propagation delay, and Doppler frequency of the lth distinguishable path.

1.2.2 Impact of physical phenomena on the propagation channel

The preceding subsections showed that the propagation channel can admit a multipath, ray-based representation, and that waves propagating along each ray may interact with the environment in various ways, described by different physical phenomena. It is proposed to evaluate what are the different impact of Doppler shifts and multipath propagation on the channel properties.

Doppler shift and time fading

As shown earlier, when physical objects are in motion within the propagation environment, a shift in the received signal frequency arises due to the relative mobility between the transmitter, receiver, and scatterers. This phenomenon, known as mobility-induced Doppler shift [5], leads to frequency dispersion and time-selective fading.

As an example, considering a moving transmitter emitting a signal $x(t)$ with frequency f_0, the frequency observed at the receiver when considering a pure LoS channel reads as $f = (1 + \Delta v/c) f_0$, with Δv representing the relative speed difference between the transmitter and receiver [6]. This frequency shift induces a time-varying phase rotation, so that the channel impulse response can be rewritten as:

$$h(t, \tau) = \alpha_{\mathrm{LoS}}(t) \, \mathrm{e}^{-\mathrm{j}2\pi f_d t} \delta(\tau - \tau_{\mathrm{LoS}}(t)), \tag{1.8}$$

where:

$$f_d = \frac{\Delta v}{c} f_0 = \frac{\Delta v}{\lambda_0}, \tag{1.9}$$

5. Each mobility source induces its own Doppler shift: pedestrians, cars, or even moving tree leaves.
6. As presented in Highlight 1.4.

is the Doppler shift associated to the LoS path. The received signal can thus be expressed as:

$$y(t,\tau) = x(t) \circledast h(t,\tau) \tag{1.10}$$
$$= \alpha_{\mathrm{LoS}}(t)\, \mathrm{e}^{-\mathrm{j}2\pi f_d t} x(t - \tau_{\mathrm{LoS}}(t)). \tag{1.11}$$

This expression clearly shows that the received signal amplitude depends on the combination between the propagation delay and the Doppler-induced phase shift. This interaction leads to constructive or destructive interference, resulting in time-selective fading of the received amplitude. From a frequency-domain perspective, taking the Fourier transform of the received signal with respect to the delay yields:

$$y(t,f) = x(t)\, \alpha_{\mathrm{LoS}}(t)\, \mathrm{e}^{-\mathrm{j}2\pi f_d t} \mathrm{e}^{-\mathrm{j}2\pi f \tau_{\mathrm{LoS}}(t)}. \tag{1.12}$$

Assuming $x(t) = \mathrm{e}^{-\mathrm{j}2\pi f_0 t}$, which respects the aforementioned assumption that the transmitted signal had frequency f_0, one obtains:

$$y(t,f) = \alpha_{\mathrm{LoS}}(t)\, \mathrm{e}^{-\mathrm{j}2\pi (f_0+f_d) t} \mathrm{e}^{-\mathrm{j}2\pi f \tau_{\mathrm{LoS}}(t)} \tag{1.13}$$
$$= \alpha_{\mathrm{LoS}}(t)\, \mathrm{e}^{-\mathrm{j}2\pi \left(f_0 + \frac{\Delta v}{\lambda_0}\right) t} \mathrm{e}^{-\mathrm{j}2\pi f \tau_{\mathrm{LoS}}(t)}, \tag{1.14}$$

where the frequency dispersion induced by the Doppler shift becomes evident. Since each moving object in the environment contributes to a distinct Doppler component, the overall frequency dispersion increases in multipath scenarios with high mobility: e.g. on a highway, each moving vehicle may generate a reflected path with its own Doppler shift. This illustrated in Fig. 1.9 and Fig. 1.10, where the first depicts different mobility scenarios, and the second shows the time evolution of a channel coefficient for each configuration. As expected, Fig. 1.10 reveals that increasing the transmitter speed, and hence the Doppler shift, enhances the time-selective fading.

Highlight 1.5: Time-selective fading

The presence of moving objects in the propagation environment introduces Doppler shifts, rendering the channel time-varying as the set of distinguishable propagation paths may evolve over time. This leads to time-selective fading, where certain scene configurations are more favorable to signal propagation more than others, as the Doppler-induced phase shifts can cause constructive or destructive interference in the received signal. In the frequency domain, these shifts result in frequency spreading, with the received signal frequency shifted proportionally to the relative velocity of each scattering object. This can be summarized as follows:

Mobility $\Leftrightarrow$ *Frequency dispersion* $\Leftrightarrow$ *Time-selective fading*

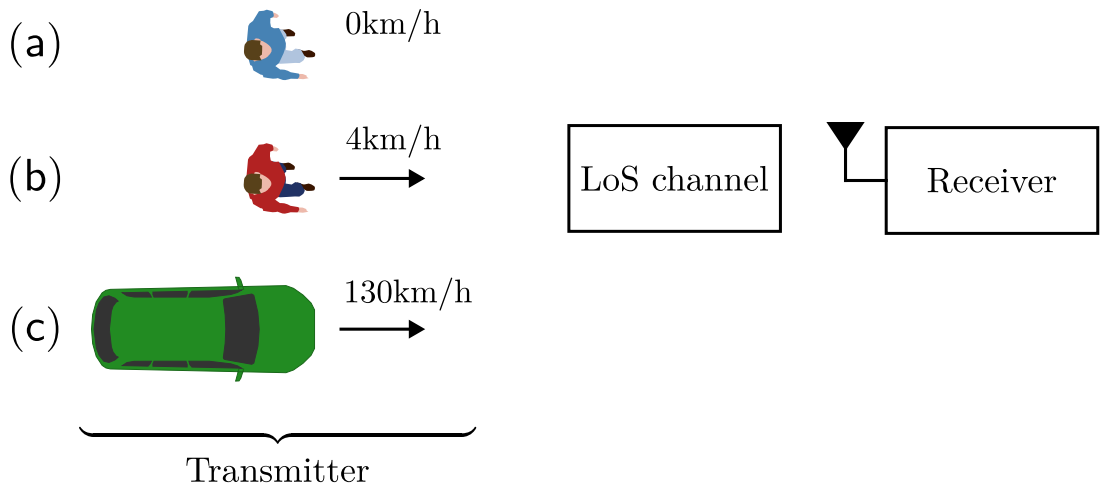


Figure 1.9: Considered mobility scenarios: in (a) the transmitter is a standing pedestrian, in (b) the transmitter is a pedestrian walking at 4 km/h, while in (c) it is a car moving at 130 km/h.

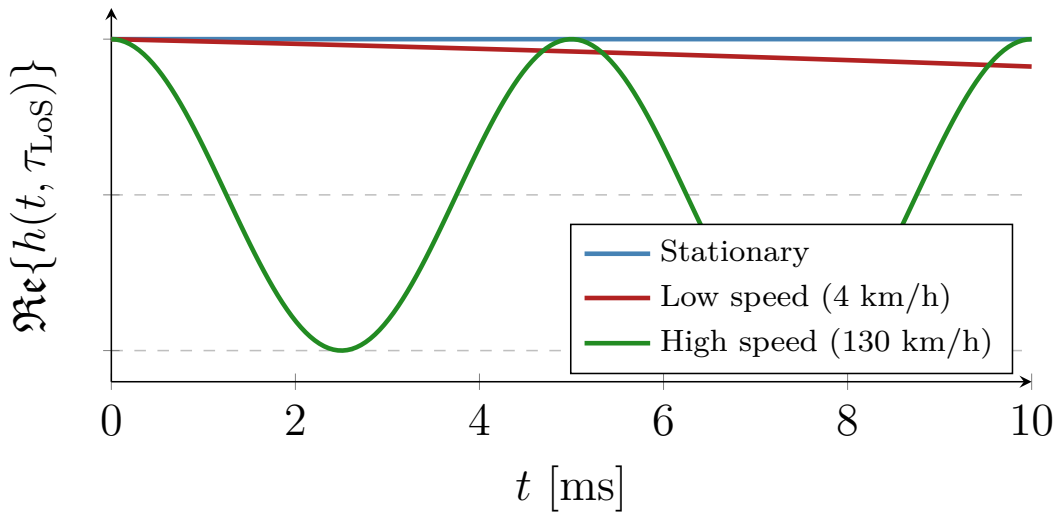


Figure 1.10: Evolution of a channel coefficient over time for different mobility scenarios.

Multipath propagation and frequency-selective fading

When multiple propagation paths are present, the received signal consists of several attenuated, delayed, and phase-shifted replicas of the transmitted signal. This can be seen using the previously defined channel impulse response: for a given signal $x(t)$ transmitted over a channel $h(t,\tau)$, the received signal $y(t,\tau)$ can be expressed as:

$$y(t,\tau) = x(t) \circledast h(t,\tau) \tag{1.15}$$

$$= \sum_{l=1}^{L_p} \alpha_l(t)\, x(t-\tau_l(t))\, \mathrm{e}^{-\mathrm{j}2\pi f_{d,l} t}. \tag{1.16}$$

One can observe in Eq. (1.16) that the amplitude, and therefore the power, of the received signal depends on the superposition of the delayed signals. This superposition can lead to either constructive or destructive interference, depending on the relative phases induced by each path. The manifestation of the multipath phenomenon in the frequency domain reads as:

$$y(t,f) = \sum_{l=1}^{L_p} \alpha_l(t)\, x(t)\, \mathrm{e}^{-\mathrm{j}2\pi f \tau_l(t)} \mathrm{e}^{-\mathrm{j}2\pi f_{d,l} t}. \tag{1.17}$$

It can be seen in Eq. (1.17) that the amplitude of the received signal spectrum depends on:

1. The attenuation coefficients $\alpha_l(t)$ induced by each propagation path.
2. The product of the considered frequency and propagation delay: $f\tau_l(t)$, as well as the product of Doppler frequency and considered time $f_{d,l}t$, both govern whether the resulting superposition is constructive or destructive. This phenomenon gives rise to frequency-selective fading.

A manifestation of the frequency-selective fading due to multipath propagation is illustrated in Fig. 1.11 and Fig. 1.12:

- In Fig. 1.11, the channel impulse responses of two different scenarios are depicted: the first considers only one path, while the second exhibits a large number of distinguishable propagation paths.
- Fig. 1.12 shows the associated channel frequency responses. In the monopath configuration, the channel is said to be frequency-flat, as its frequency response remains constant across all the frequency domain. In contrast, the multipath case clearly reveals the frequency-selective nature of the channel, characterized by multiple attenuation peaks at specific frequencies, resulting from destructive interference between delayed signal components.

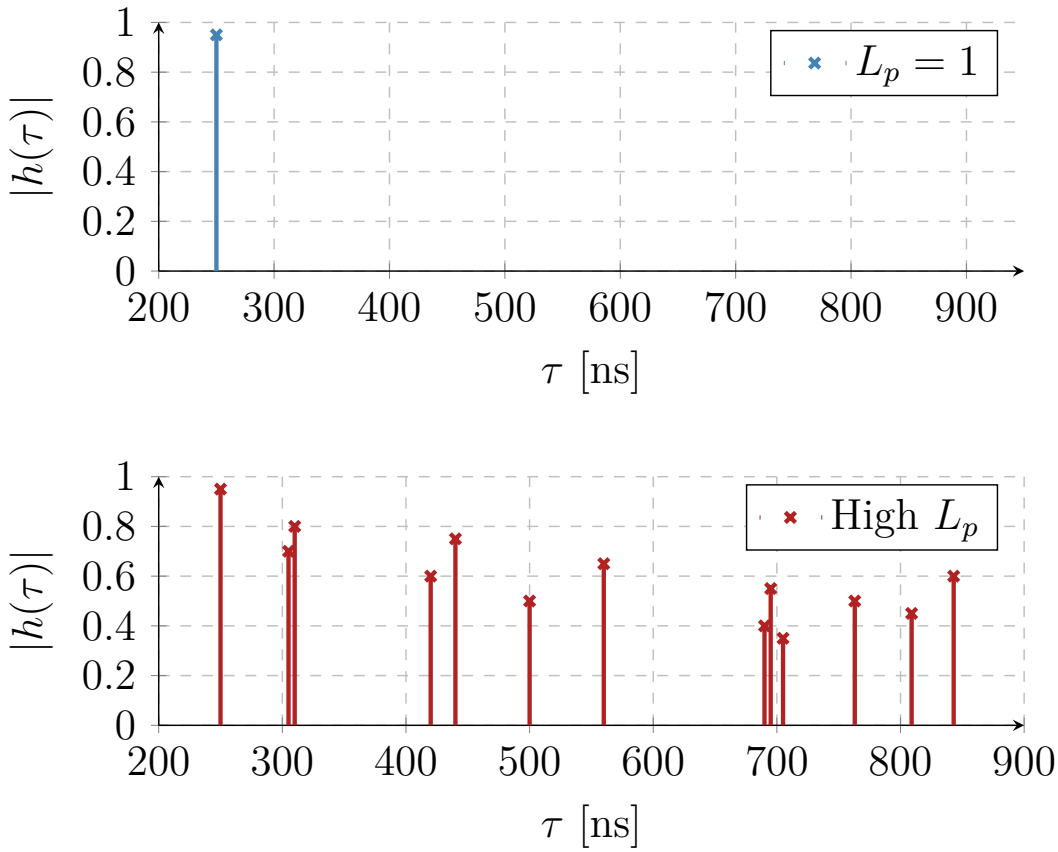


Figure 1.11: Channel impulse responses for two path configurations.

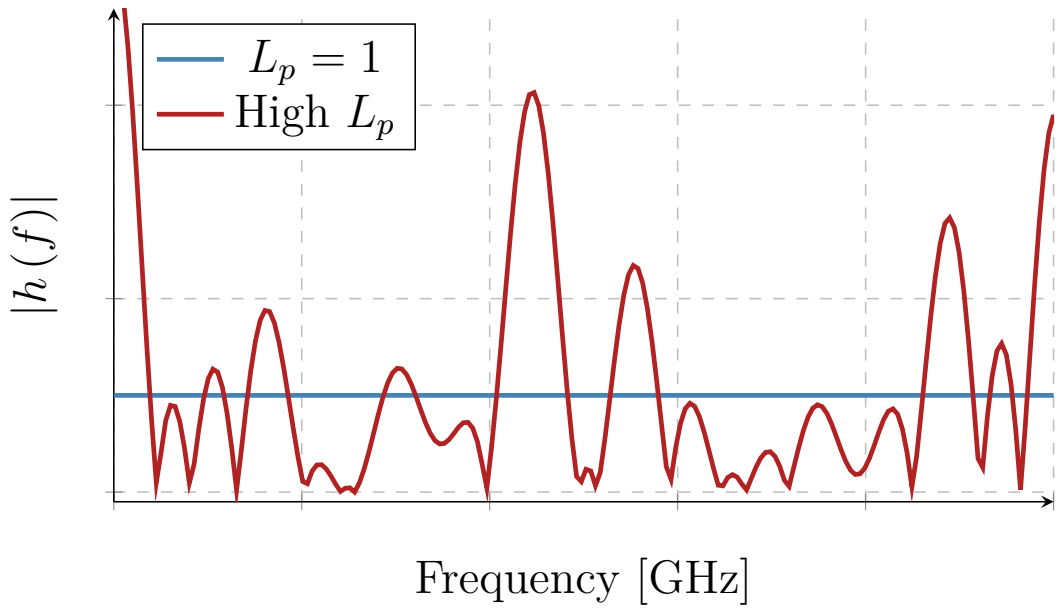


Figure 1.12: Channel frequency responses for two path configurations.

Highlight 1.6: Frequency-selective fading

The presence of multiple propagation paths induces time dispersion of the transmitted signal, as the different components arrive at the receiver with varying delays. The superposition of these components can result in either constructive or destructive interference, leading to frequency-selective fading, manifested as amplitude variations in the received signal's spectrum. This can be summarized as follows:

Multipath propagation $\Leftrightarrow$ *Time dispersion* $\Leftrightarrow$ *Frequency-selective fading*

Notions of coherence bandwidth/time

As multipath propagation and mobility induce both frequency selectivity and time variability in the wireless channel, one may naturally ask whether there exists a frequency range and a time interval over which the channel remains approximately invariant. This question motivates the introduction of the coherence bandwidth and coherence time concepts, which respectively quantify the frequency and time stability of the channel. These quantities are evaluated from a statistical perspective, using the time-frequency channel autocorrelation function.

Definition 1.3: Time-frequency autocorrelation function

Given a channel frequency response $h(t, f)$, under the assumption that the fast variations of the channel in the time domain are wide-sense stationary and that the attenuations across distinct propagation paths are uncorrelated, the time-frequency autocorrelation function is defined as:

$$\Gamma_h(\Delta t, \Delta f) = \int_{\mathbb{R}} \int_{\mathbb{R}} h(t, f)\, h^*(t - \Delta t, f - \Delta f)\, \mathrm{d}t\mathrm{d}f. \tag{1.18}$$

The coherence bandwidth at $\kappa_f\%$, $B^c_{\kappa_f}$ defines the minimum frequency spacing so that the fading experienced at those frequencies is decorrelated by $\kappa_f\%$. In other words, given two distinct frequencies $(f_1, f_2) \in \mathbb{R}^2$, if $|f_1 - f_2| > B^c_{\kappa_f}$, the channel responses at f_1 and f_2 are considered statistically uncorrelated at the $\kappa_f\%$ level. A formal definition of the coherence bandwidth is presented below and is illustrated in Fig. 1.13 for $\kappa_f = 50\%$.

Definition 1.4: Coherence bandwidth

Given the time-frequency response $\Gamma_h(\Delta t, \Delta f)$, the coherence bandwidth at $\kappa_f\%$, $B^c_{\kappa_f}$, is defined such that:

$$\kappa_f = \mathbb{E}_{\Delta t}\left[\frac{\left|\Gamma_h\left(\Delta t, B^c_{\kappa_f}\right)\right|}{\left|\Gamma_h(\Delta t, 0)\right|}\right]. \tag{1.19}$$

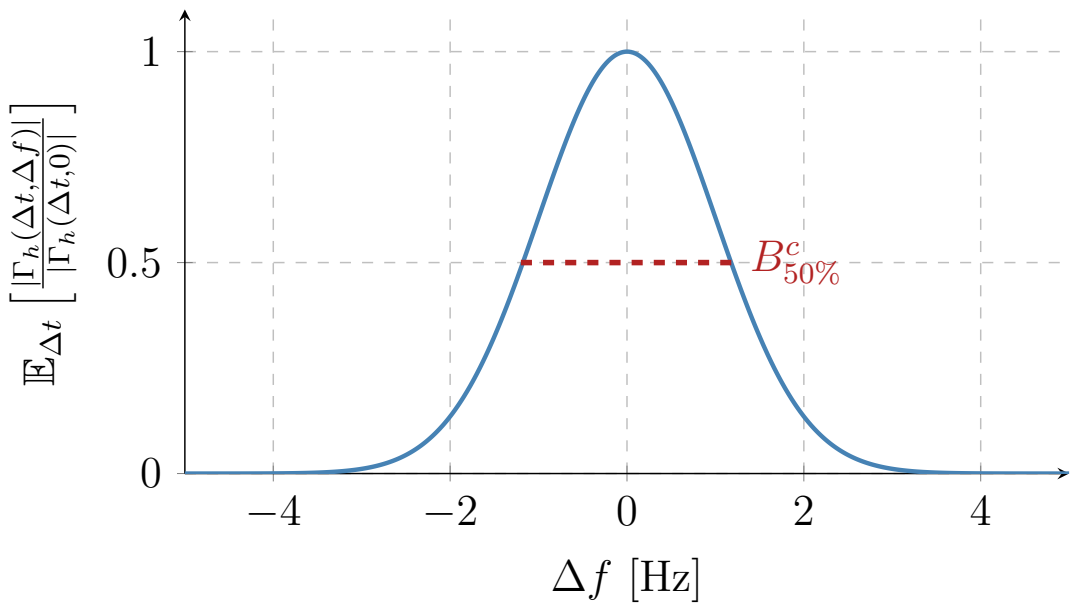


Figure 1.13: Illustration of the coherence bandwidth at 50%, for an arbitrary channel.

It can be shown that the coherence bandwidth is inversely proportional to the standard deviation of the propagation delays, denoted as σ_τ. As the number of distinguishable propagation paths increases, σ_τ becomes larger, leading to a smaller coherence bandwidth. This means that statistically, frequency-selective fading will occur more frequently in highly multipath environments. This is consistent with Highlight 1.6: rich multipath environments tend to exhibit stronger frequency selectivity.

An analogous quantity is defined over the time domain: the coherence time at $\kappa_t\%$, denoted as $T^c_{\kappa_t}$. It defines the minimum time spacing so that two time-selective fading are uncorrelated by $\kappa_t\%$. A formal definition is proposed in Definition 1.5, and an illustration for $\kappa_t = 50\%$ is presented in Fig. 1.14.

Definition 1.5: Coherence time

Given the time-frequency response $\Gamma_h(\Delta t, \Delta f)$, the coherence time at κ_t%, $T^c_{\kappa_t}$, is defined such that:

$$\kappa_t = \mathbb{E}_{\Delta f}\left[\frac{\left|\Gamma_h\left(T^c_{\kappa_t}, \Delta f\right)\right|}{\left|\Gamma_h(0, \Delta f)\right|}\right]. \tag{1.20}$$

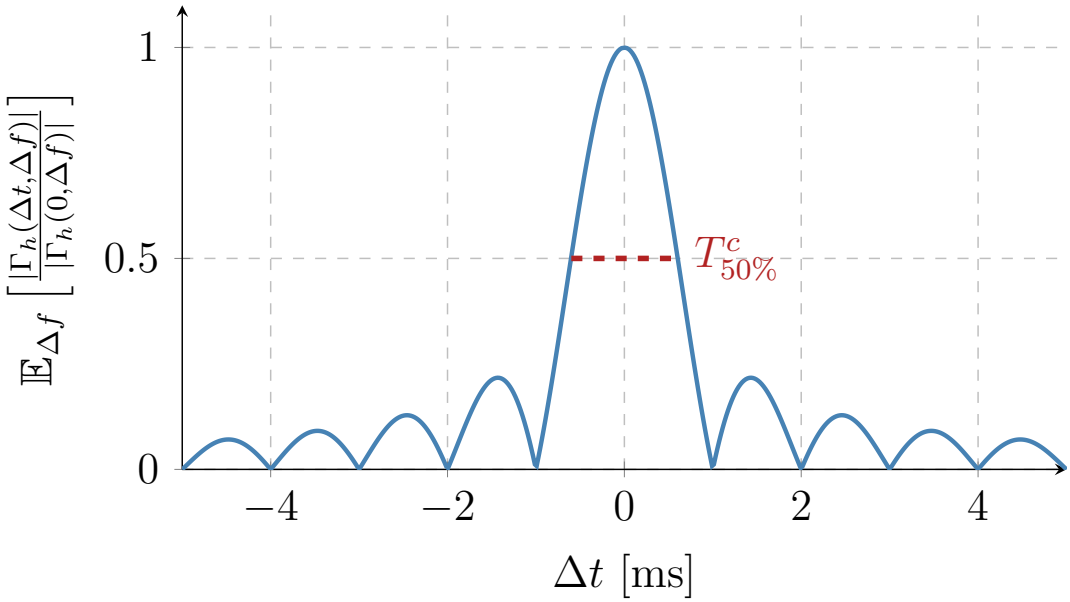


Figure 1.14: Illustration of the coherence time at 50%, for an arbitrary channel.

The coherence time is inversely proportional with the maximum Doppler frequency in the channel, which is consistent with Highlight 1.6: a propagation channel with high mobility, i.e. high values of f_d, is expected to present very strong time-selective fading. In such cases, the channel characteristics fluctuate rapidly over time, meaning that the duration between two distinct fading events is short, and thus the coherence time is low.

To conclude on these notions, since the coherence time defines the interval over which the channel can be considered statistically constant, the channel is typically assumed to be time-invariant within time blocks of duration equal to the coherence time. In an similar manner, the coherence bandwidth defines the frequency range over which the channel exhibits decorrelated frequency fading, which means that the channel can be assumed to be frequency-flat within frequency bands of width equal to the coherence bandwidth.

1.3 Application to modern communication systems

Having introduced the fundamentals of propagation channel modeling, a natural question arises: how can these channel models be leveraged to enhance the performance of wireless communication systems?

1.3.1 Channel capacity

Before introducing the role that the propagation channel plays in modern communication systems, it is primordial to establish a notion of performance limits. Indeed, when seeking to improve the performance of a wireless transmission system, it is natural to ask whether further improvements are theoretically possible. The tool used to determine these fundamental performance limits is known as the channel capacity, a concept established in the aforementioned Shannon paper [Sha48], which laid the foundation of information theory.

Definition 1.6: Channel capacity

The channel capacity is defined as the maximum rate at which information can be transmitted over a communication channel with an arbitrarily small transmission error probability. It is usually expressed in bit per second (bps).

As an example, the Shannon-Hartley theorem states that, for a channel affected by additive white Gaussian noise (AWGN), with bandwidth B, and signal-to-noise ratio (SNR), the channel capacity C is given by:

$$\mathsf{C} = B \log_2 (1 + \mathsf{SNR}) . \tag{1.21}$$

The SNR being defined as the received signal power over the channel noise power, it becomes clear that the channel plays a predominant role in this theoretical bound:

1. The channel can attenuate the received signal, reducing its amplitude and power, thereby lowering the SNR and decreasing channel capacity.

2. Channels with high noise levels increase noise power, thereby reducing the SNR and thus the channel capacity.

An illustrative example of the channel impact on this performance bound can be obtained with a simple LoS AWGN channel with no mobility. In this setting, the received signal $y(t)$ can be expressed as:

$$y(t) = \alpha_{\mathrm{LoS}} x (t - \tau_{\mathrm{LoS}}(t)) + n(t) , \tag{1.22}$$

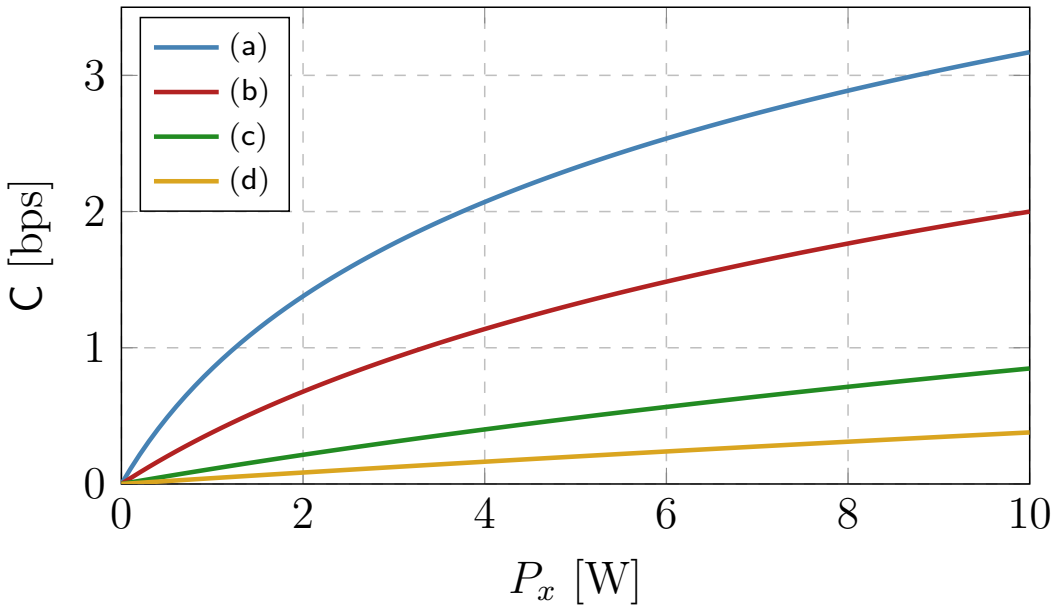


Figure 1.15: Evolution of the channel capacity of a LoS channel, for different channel attenuation/noise configurations: for this example, $B = 1$Hz. (a): $(\sigma_n^2 = 1, |\alpha_l|^2 = 0.8)$, low noise/low attenuation. (b): $(\sigma_n^2 = 1, |\alpha_l|^2 = 0.3)$, low noise/high attenuation. (c): $(\sigma_n^2 = 10, |\alpha_l|^2 = 0.8)$, high noise/low attenuation. (d): $(\sigma_n^2 = 10, |\alpha_l|^2 = 0.3)$, high noise/high attenuation.

where $n(t) \sim \mathcal{CN}(0, \sigma_n^2)$ denotes the noise, and $x(t)$ is the transmitted signal with bandwidth B and power P_x. The SNR reads as:

$$\mathsf{SNR} = \frac{|\alpha_{\text{LoS}}|^2 P_x}{\sigma_n^2}, \tag{1.23}$$

leading to the following channel capacity expression:

$$\mathsf{C} = B \log_2 \left(1 + \frac{|\alpha_{\text{LoS}}|^2 P_x}{\sigma_n^2} \right), \tag{1.24}$$

which is a decreasing function of the noise power σ_n^2, and an increasing function of the channel attenuation amplitude $|\alpha_{\text{LoS}}|^2$ and signal power P_x. The behavior of Eq. (1.24) is illustrated in Fig. 1.15, where plots for different noise and channel powers show the dependence of the capacity on the different channel properties.

1.3.2 Multi-antenna systems

Thus far, the transmitter system has been considered to employ a single antenna. In modern communication systems, however, both the transmitter and the receiver are typically

equipped with multiple antennas, introducing an additional degree of freedom: the spatial domain. This enables advanced techniques such as beamforming and spatial multiplexing. Such antenna configurations are referred to as multiple-input multiple-output (MIMO) systems.

Mathematical model of a MIMO channel

In case of a MIMO system equipped with N_t transmit antennas and N_r receive antennas, the received signal becomes a time-dependent vector $\mathbf{y}(t,\tau) \in \mathbb{C}^{N_r}$, which contains what is received by each receive antenna. For an emitted signal $\mathbf{x}(t)$, the expression of $\mathbf{y}(t,\tau)$ reads as:

$$\mathbf{y}(t,\tau) = \begin{bmatrix} y_1(t,\tau) \\ \vdots \\ y_{N_r}(t,\tau) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N_t} x_i(t) \circledast h_{1,i}(t,\tau) \\ \vdots \\ \sum_{i=1}^{N_t} x_i(t) \circledast h_{N_r,i}(t,\tau) \end{bmatrix}, \tag{1.25}$$

with $h_{j,i}(t,\tau)$ representing the channel impulse response between the jth receiver and ith transmitter antenna. In the frequency domain, Eq. (1.25) simplifies as the convenient matrix product:

$$\mathbf{y}(t,f) = \mathbf{H}(t,f)\,\mathbf{x}(t), \tag{1.26}$$

with $\mathbf{H}(t,f) \in \mathbb{C}^{N_r \times N_t}$ defined as the *channel matrix*, i.e. the channel frequency responses over each transmit/receive antenna pairs.

A well-established model for the MIMO channel is based on the far-field assumption. When the transmitter-receiver distance exceeds the Fraunhofer distance [SJ17; CD22], the channel can be accurately described using a ray-based model [7]. This approach, which assumes planar wavefronts impinging on the antenna arrays, forms the basis of Definition 1.7, introducing the steering vector, and Definition 1.8, which employs steering vectors in the construction of a ray-based channel model. A steering vector describes the phase shifts induced across the antenna array by an incoming wavefront arriving at a given angle relative to the array normal. These phase shifts arise from the differences in propagation distance between the wavefront and each antenna element, as illustrated in Fig. 1.16.

7. See Section 1.2.1, [TV05, Chapter 7, pp. 311-316] or [BD24, Chapter 5, pp. 385-386].

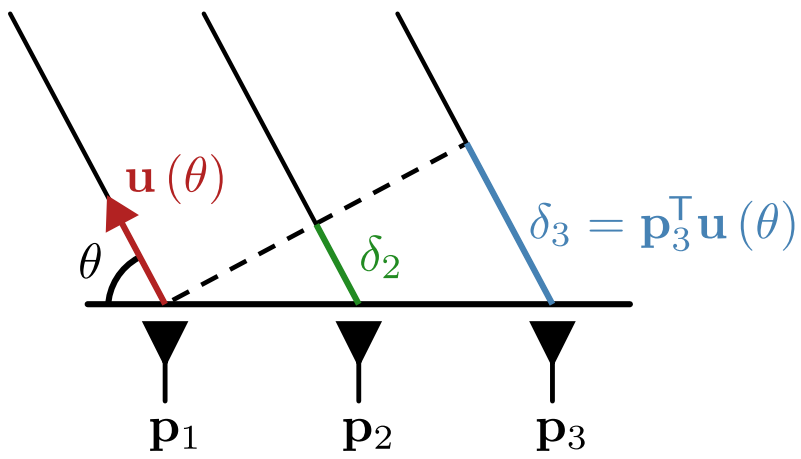


Figure 1.16: Illustration of the path difference for a wavefront impinging on a 1D antenna array at azimuth angle θ. The path difference $\delta_i = \mathbf{p}_i^\mathsf{T}\mathbf{u}(\theta)$ induces a phase shift in the received signal, leading to the well-known steering vector model.

Definition 1.7: Steering vectors

Given a multi-antenna system with N distinct antenna elements located at positions $\mathbf{p}_i \in \mathbb{R}^3$, the steering vector associated with an impinging plane wave arriving from azimuth/elevation angles (θ, ϕ) is given by:

$$\mathbf{a}(\theta, \phi) = \frac{1}{\|\mathbf{g}\|_2} \begin{bmatrix} g_1 \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda}\mathbf{p}_1^\mathsf{T}\mathbf{u}(\theta,\phi)} \\ \vdots \\ g_N \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda}\mathbf{p}_N^\mathsf{T}\mathbf{u}(\theta,\phi)} \end{bmatrix}, \tag{1.27}$$

where $\mathbf{g} \in \mathbb{C}^N$ denotes the complex gain of each antenna elements, λ denotes the considered wavelength, and $\mathbf{u}(\theta, \phi) \in \mathbb{R}^3$ denotes a unit norm vector in the direction (θ, ϕ).

Definition 1.8: MIMO channel frequency response

Given a propagation channel with L_p distinguishable propagation paths, and assuming that the far-field assumption holds, the channel frequency response is given by:

$$\mathbf{H}(t,f) = \sum_{l=1}^{L_p} \alpha_l(t)\, \mathrm{e}^{-\mathrm{j}\left(2\pi f \tau_l(t) + \frac{\Delta v_l}{\lambda_0} t\right)} \mathbf{a}_{\mathrm{r}}\left(\theta_l^{\mathrm{a}}, \phi_l^{\mathrm{a}}\right) \mathbf{a}_{\mathrm{t}}^{\mathsf{H}}\left(\theta_l^{\mathrm{d}}, \phi_l^{\mathrm{d}}\right), \tag{1.28}$$

where $\alpha_l(t) \in \mathbb{C}$ is the complex attenuation, $\tau_l(t)$ the propagation delay, and Δv_l the relative transmitter/receiver speed for the lth path; λ_0 denotes the wavelength corresponding to the system's central frequency; $(\theta_l^{\mathrm{a}}, \phi_l^{\mathrm{a}})$, resp. $\left(\theta_l^{\mathrm{d}}, \phi_l^{\mathrm{d}}\right)$ represent the azimuth and elevation angles of arrival, resp. departure, for the lth path. Additionally, $\mathbf{a}_{\mathrm{t}}(\theta, \phi)$ and $\mathbf{a}_{\mathrm{r}}(\theta, \phi)$ represent the transmitter/receiver steering vectors.

It should be noted that the MIMO channel model presented in Definition 1.8 can be extended when the far-field assumption does not hold by considering near-field steering vectors [HB91; Fri19; LD23; Gon+24; GLMS25].

Similarly to the mono-antenna scenario, now denoted as single input single output (SISO), the communication performance over a MIMO channel are bounded by the capacity, expressed by Telatar [Tel99] as follows.

Definition 1.9: MIMO channel capacity

Given a MIMO channel, of channel matrix $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$, considering AWGN of power σ_n^2, bandwidth B, the MIMO capacity can be expressed as:[a]

$$\mathsf{C}_{\mathrm{MIMO}} = B \log_2\left(\det\left(\mathbf{Id}_{N_r} + \frac{1}{\sigma_n^2}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{H}}\right)\right), \tag{1.29}$$

where $\mathbf{Q} = \mathbb{C}^{N_t \times N_t}$ is a power allocation matrix.

a. Note that the time/frequency dependence of the channel matrix is omitted here.

Assuming uniform power allocation, the capacity of a MIMO channel can be expressed as the sum of orthogonal SISO channels capacities, each corresponding to an eigenmode of the MIMO channel matrix. This decomposition is obtained through the singular value decomposition (SVD) of the channel matrix, which diagonalizes the MIMO channel into a set of independent subchannels, each associated with an eigenvalue value that determines

its individual capacity contribution [Tel99]. This is formalized as:

$$\mathsf{C}_{\text{MIMO}} = B \sum_{i=1}^{\text{rk}(\mathbf{H})} \log_2 \left(1 + \frac{P_e}{N_t \sigma_n^2} \lambda_i \right), \tag{1.30}$$

P_e being the total emitted power, i.e. $\text{Tr}(\mathbf{Q})$, and λ_i being the ith eigenvalue of $\mathbf{H}\mathbf{H}^{\mathsf{H}}$, which is the square of the ith singular value of $\mathbf{H}$. A demonstration of Eq. (1.30) is proposed in Appendix A.3.

Under the assumption of uniform power allocation, and denoting the ith orthogonal SISO sub-channel capacity as $\mathsf{C}_{\text{SISO}_i}$, it then clearly appears that:

$$\forall i \in [\![1, \text{rk}(\mathbf{H})]\!],\ \mathsf{C}_{\text{MIMO}} > \mathsf{C}_{\text{SISO}_i}. \tag{1.31}$$

It is important to note that although multi-antenna systems offer the potential for increased capacity over mono-antenna systems, their transmission performance is still fundamentally constrained by the channel properties. As it is stated above, in the best-case scenario, the system can transmit simultaneously over $\text{rk}(\mathbf{H})$ distinct SISO channels, corresponding to the eigenmodes of the MIMO channel. However, if the channel matrix is poorly conditioned, i.e. if it exhibits small or highly unequal eigenvalues, the effective number of usable distinct eigenmodes is reduced, leading to degraded transmission performance.

MIMO performance optimization

Over the past two decades, numerous techniques have been developed to mitigate the potential performance degradation introduced by the MIMO propagation channel. It is proposed to focus on two of them: water-filling and beamforming.

From a theoretical perspective, optimizing transmission performance amounts to maximizing the effective channel capacity, which can be achieved by appropriately allocating the transmit power across the transmit antennas. This formally reads as the following optimization problem:

$$\begin{aligned} \mathsf{C}^{\star}_{\text{MIMO}} = \max_{\mathbf{Q} \in \mathbb{R}^{N_t \times N_t}} \quad & B \log_2 \left(\det \left(\mathbf{Id}_{N_r} + \frac{1}{\sigma_n^2} \mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{H}} \right) \right), \\ \text{s.t.} \quad & \text{Tr}(\mathbf{Q}) \leq P_{\max} \end{aligned} \tag{1.32}$$

where the constraint limits the total transmitted power. Note that in order to solve the problem described in Eq. (1.32), full channel knowledge is required at the transmitter. This problem is convex and can be solved using Lagrangian methods [8]. In that case, given the set of active antennas $\mathcal{A} = \{1, \cdots, \text{rk}(\mathbf{H})\}$, the allocated power p_i to the ith

8. An interested reader is referred to [BD24, Chapter 3, pp.175-177], or [TV05, Chapter 5, pp. 182-184].

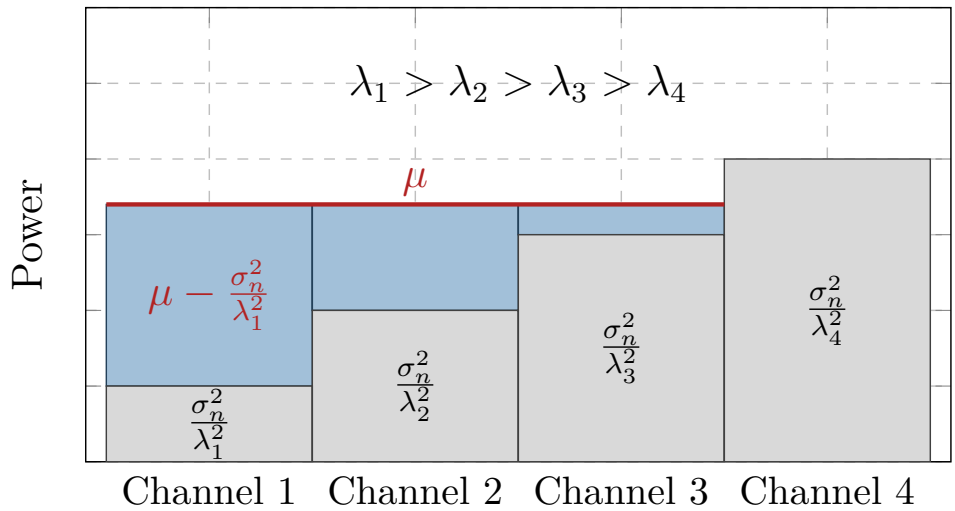


Figure 1.17: Illustration of the waterfilling power allocation method (based on Fig. 3.12 in [BD24]).

transmitter antenna is:

$$\begin{cases} \forall i \in \mathcal{A}, & p_i = \max\left(\mu - \frac{\sigma_n^2}{\lambda_i}, 0\right) \\ \forall i \notin \mathcal{A}, & p_i = 0 \end{cases}, \tag{1.33}$$

where μ is an optimization variable taken so that $\sum_{i=1}^{\mathrm{rk}(\mathbf{H})} p_i = P_{\max}$. The associated optimal power allocation matrix [9] is $\mathbf{Q}^\star = \mathbf{V}\mathrm{diag}\left(p_1, \cdots, p_{\mathrm{rk}(\mathbf{H})}, \mathbf{0}^\mathsf{T}_{\mathbb{R}^{N_t - \mathrm{rk}(\mathbf{H})}}\right)\mathbf{V}^\mathsf{H}$, where $\mathbf{V}$ originates from the SVD of $\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\mathsf{H}$. The key insight from Eq. (1.33) is that each eigenmode subchannel is assigned a distinct power level, which depends both on the total power budget $P_{\max}$, through μ, and on each subchannel quality, through the eigenvalues λ_i. This optimal power allocation policy follows the water-filling principle: more power is allocated to the stronger eigenmodes, while weaker modes may receive little to no power, a principle which is illustrated in Fig. 1.17. Under this approach, favorable subchannels, i.e. high λ_i, contribute more significantly to the overall capacity through larger power allocations.

In addition to optimizing the power allocation policy, another approach to enhance the effective channel capacity is to focus the transmitted power along specific spatial directions, a technique known as SVD beamforming [BD24, Chapter 3, pp. 185-197]. Such approach requires both the transmitter and receiver to have full channel knowledge, as the selected spatial directions consists of the channel matrix singular vectors [10]. From a more technical perspective, the general beamforming technique relies on applying specific phase shifts across the transmit and receive antenna elements, such that the resulting

9. If $N_t < N_r$, the diagonal of $\mathbf{Q}$ does not contain null values, i.e. all transmit antennas are used.
10. Note that this assumption entails significant complexity as a specific channel estimation procedure is required to obtain channel knowledge at both the transmitter and receiver level.

electromagnetic wavefronts constructively interfere in a desired direction. This enables the transmitter to steer the radiated energy towards a specific spatial direction, or the receiver to enhance signals arriving from a given direction. Formally, both a precoder $\mathbf{W} \in \mathbb{C}^{N_t \times N_t}$ and combiner $\mathbf{F} \in \mathbb{C}^{N_r \times N_r}$ are defined, so that Eq. (1.26) can be rewritten as follows:

$$\mathbf{y}(t,f) = \mathbf{F}(t,f)\,\mathbf{H}(t,f)\,\mathbf{W}(t,f)\,\mathbf{x}(t)\,. \tag{1.34}$$

For the sake of clarity in the following developments, the frequency dependence will be omitted, and the time index will be assumed fixed, leading to:

$$\mathbf{y} = \mathbf{FHWx}. \tag{1.35}$$

Using the SVD approach, it can be shown that the channel matrix admits the following representation:

$$\mathbf{H} = \mathbf{U\Sigma V}^{\mathsf{H}} \tag{1.36}$$

$$= \sum_{i=1}^{\mathrm{rk}(\mathbf{H})} \sqrt{\lambda_k}\mathbf{u}_i\mathbf{v}_i^{\mathsf{H}}, \tag{1.37}$$

with $\mathbf{U} \in \mathbb{C}^{N_r \times N_r}$, $\mathbf{V} \in \mathbb{C}^{N_t \times N_t}$ being unitary, i.e. $\mathbf{U}^{\mathsf{H}}\mathbf{U} = \mathbf{Id}_{N_r}$ $\mathbf{V}^{\mathsf{H}}\mathbf{V} = \mathbf{Id}_{N_t}$, and $\mathbf{\Sigma} \in \mathbb{R}^{N_r \times N_t}$ being rectangular diagonal. Eq. (1.37) highlights the channel eigenmodes: it reveals that the MIMO channel offers $\mathrm{rk}(\mathbf{H})$ orthogonal spatial paths where $\mathbf{v}_i$ and $\mathbf{u}_i$ respectively describe the transmit and receive spatial directions. An illustration of this behavior is proposed in Fig. 1.18. From this representation, one can quickly show that taking:

$$\mathbf{W} = \mathbf{V} \tag{1.38}$$

$$\mathbf{F} = \mathbf{U}^{\mathsf{H}}, \tag{1.39}$$

i.e. aligning the transmitted and received wavefronts with the channel's eigenmode directions, is optimal from a capacity standpoint. Indeed, in that case, the received signal rewrites as:

$$\mathbf{y} = \mathbf{U}^{\mathsf{H}}\mathbf{U\Sigma V}^{\mathsf{H}}\mathbf{Vx} \tag{1.40}$$

$$= \mathbf{\Sigma x} \tag{1.41}$$

$$= \left[\sqrt{\lambda_1}x_1 \quad \cdots \quad \sqrt{\lambda_{\mathrm{rk}(\mathbf{H})}}x_{\mathrm{rk}(\mathbf{H})} \quad 0 \quad \cdots \quad 0\right]^{\mathsf{T}}, \tag{1.42}$$

which translates to $\mathrm{rk}(\mathbf{H})$ orthogonal SISO links, each associated with a channel eigenmode. Assuming uniform power allocation across all transmit antennas, and that the

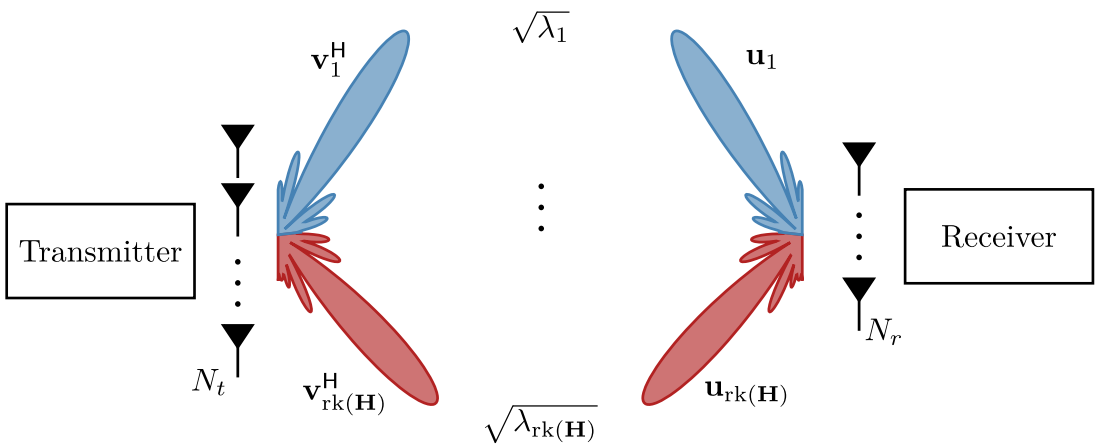


Figure 1.18: Illustration of the channel matrix eigenmodes (based on Fig. 3.16 in [BD24]).

channel suffers from AWGN with power σ_n^2, the capacity reads as:

$$\mathsf{C} = B \sum_{i=1}^{\mathrm{rk}(\mathbf{H})} \log_2 \left(1 + \frac{P_e}{N_t \sigma_n^2} \lambda_i\right), \tag{1.43}$$

which is Eq. (1.30): this confirms the optimality of this particular beamforming scheme with uniform power allocation.

Remark 1.2. *When considering a MIMO multipath channel with a number of propagation path smaller than the number of transmit and receive antennas, i.e. $N_t > L_p$, $N_r > L_p$, and angularly sufficiently separated propagation paths so that the corresponding steering vectors are (almost) orthogonal, each propagation path corresponds to a distinct channel eigenmode, as illustrated in Fig. 1.18. However, in the general case, each of the channel eigenmode arises from a linear combination of multiple propagation paths, restricting the visual intuition of the eigenmode concept.*

Application to multi-user transmission

In a wireless cellular scenario, a fixed transmitter, referred to as a base station (BS), is tasked to serve multiple user equipments (UEs), which are potentially moving. When the BS is equipped with multiple antennas, the scenario is referred to as multi-user MIMO (MU-MIMO). This setting gives rise to the concept of multiplexing, where several users share the same communication resources. Several multiplexing strategies can be envisioned, as presented below, and illustrated in Fig. 1.19.

- **Time multiplexing.** Known as time division multiple access (TDMA), it involves allocating distinct time slots to each UE, such that only one user is served at any given time. While conceptually simple, TDMA requires precise time synchronization, involves complex scheduling mechanisms, and introduces inherent latency, especially

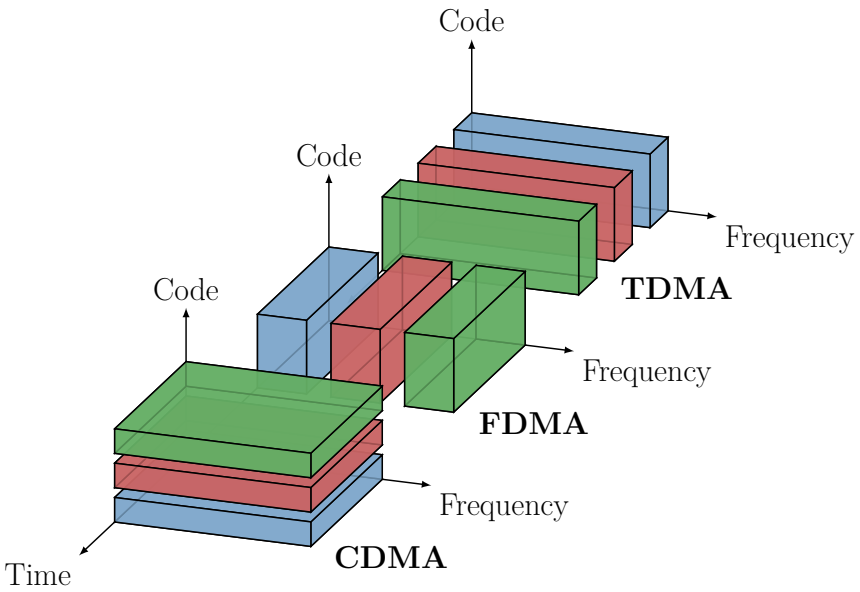


Figure 1.19: Illustration of the TDMA/FDMA/CDMA schemes for 3 distinct users: each user is represented by a different color.

in scenarios with a high number of users, which can be problematic when the system has delay constraints.

- **Code multiplexing.** Known as code division multiple access (CDMA), this approach assigns a distinct code to each UE, enabling the base station to serve all users simultaneously over the same time and frequency resources. CDMA offers efficient time and frequency use, in addition to good interference rejection capabilities. However, CDMA systems tend to be complex to implement due to power management constraints for close and far UEs, and due to the code generation, assignment and management.

- **Frequency multiplexing.** Known as frequency division multiple access (FDMA), this scheme divides the available spectrum into orthogonal frequency sub-bands, each assigned to a distinct UE. This frequency separation allows efficient interference mitigation in addition to simple implementation requirements. However, this multiplexing scheme can lead to inefficient spectrum use, especially when UEs bandwidth requirements are inferior to the allocated sub-bands size, or if sub-bands are not allocated.

Beyond the time, code, and frequency domains, the spatial dimension of the wireless channel can also be exploited for multiplexing [11]. Similarly to point-to-point MIMO systems, where wavefronts are directed along the eigenmodes of the channel, multi-user systems

11. Note that these domains are not mutually exclusive, e.g. UEs can be jointly separated in both the frequency and code domains.

can instead steer the transmitted energy towards the spatial directions associated with distinct users. This strategy, known as spatial multiplexing or spatial division multiple access (SDMA), enables the base station to serve multiple UEs simultaneously over the same time and frequency resources by leveraging the spatial separability of their channels. In contrast with point-to-point MIMO systems, MU-MIMO introduces an additional interference constraint: UE k must only receive its intended signal and not the signals intended to other UEs j, $j \neq k$. Mitigating inter-UE interference is a complex task that is described as follows.

Considering a BS with N transmit antennas, and K mono-antenna UEs, the channel between the BS and UE k is $\mathbf{h}_k \in \mathbb{C}^N$. As such, the channel matrix is defined as the concatenation of the channel corresponding to each UE [12]: $\mathbf{H}(t,f) = [\mathbf{h}_1, \cdots, \mathbf{h}_K] \in \mathbb{C}^{N\times K}$. Omitting the frequency dependence and fixing the time, for an emitted signal $\mathbf{x} \in \mathbb{C}^K$, the received signal at the UE level, $\mathbf{y} \in \mathbb{C}^K$, over the channel $\mathbf{H}$ reads as:

$$\mathbf{y} = \eta \mathbf{H}^{\mathsf{H}} \mathbf{W} \mathbf{x} + \mathbf{n}, \tag{1.44}$$

where $\mathbf{n}$ represents AWGN, i.e. $\mathbf{n} \sim \mathcal{CN}\left(\mathbf{0}_{\mathbb{C}^K}, \sigma_n^2 \mathbf{Id}_K\right)$, $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_K] \in \mathbb{C}^{N\times K}$ is the precoding matrix, and $\eta = 1/\left\|\mathbf{W}\right\|_{\mathsf{F}}$. The received signal at UE k rewrites as:

$$y_k = \eta \mathbf{h}_k^{\mathsf{H}} \mathbf{w}_k x_k + \eta \sum_{j\neq k} \mathbf{h}_k^{\mathsf{H}} \mathbf{w}_j x_j + n_k, \tag{1.45}$$

where $\sum_{j\neq k} \mathbf{h}_k^{\mathsf{H}} \mathbf{w}_j x_j$ constitutes the inter-UE interference. Indeed UE k receives its intended signal x_k but also a linear combination of the signals to the other UEs x_j. The metrics quantifying the impact of interference, as well as interference and noise on a given transmission are the signal-to-interference ratio (SIR), and the signal-to-interference-and-noise ratio (SINR), respectively. For the received signal at UE k, under the assumption of unitary signals i.e. $\mathbb{E}\left[|x_k|^2\right] = \mathbb{E}\left[|x_j|^2\right] = 1$, the SNR, SIR, and SINR read as:

$$\mathsf{SNR} = \frac{\left|\eta \mathbf{h}_k^{\mathsf{H}} \mathbf{w}_k\right|^2}{\sigma_n^2} \tag{1.46}$$

$$\mathsf{SIR} = \frac{\left|\mathbf{h}_k^{\mathsf{H}} \mathbf{w}_k\right|^2}{\sum_{j\neq k} \left|\mathbf{h}_k^{\mathsf{H}} \mathbf{w}_j\right|^2} \tag{1.47}$$

$$\mathsf{SINR} = \frac{\left|\eta \mathbf{h}_k^{\mathsf{H}} \mathbf{w}_k\right|^2}{\sigma_n^2 + \sum_{j\neq k} \left|\eta \mathbf{h}_k^{\mathsf{H}} \mathbf{w}_j\right|^2}. \tag{1.48}$$

One can observe that inter-UE interference mitigation can be realized through the use

12. Note that the channel matrix dimensions are reversed compared to the conventional point-to-point MIMO channel formulation: this is of no particular importance as an Hermitian is involved in the input-output relation presented in Eq. (1.44), thereby preserving the mathematical equivalence.

Table 1.1: Usual precoders and their associated objectives: maximum ratio transmission (MRT), equal gain transmission (EGT), zero forcing (ZF) and minimum mean squared error (MMSE).

	$\mathbf{W}$	Objective (maximization)
MRT	$\mathbf{H}$	SNR
EGT	$\exp\left(\mathrm{j}\arg\left(\mathbf{H}\right)\right)$	SNR
ZF	$\mathbf{H}\left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1}$	SIR
MMSE	$\mathbf{H}\left(\mathbf{H}^{\mathsf{H}}\mathbf{H}+\sigma_n^2\mathbf{Id}_K\right)$	SINR

of a carefully chosen precoder, as the interference level is only dependent on the sum $\sum_{j\neq k}\left|\eta\mathbf{h}_k^{\mathsf{H}}\mathbf{w}_j\right|^2$. Based on different optimization problems, several different precoder expressions have been developed: some of the most usual precoders and their respective objective are found in Table 1.1, while an illustration of the principles underlying the ZF, MRT and MMSE precoders is provided in Fig. 1.20.

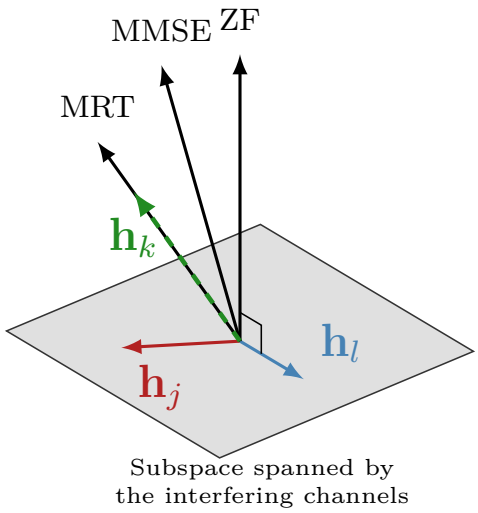


Figure 1.20: Illustration of the MRT/MMSE/ZF principle: $\mathbf{h}_k$ denotes the channel of the considered UE, while $\mathbf{h}_j$ and $\mathbf{h}_l$ correspond to the channels of interfering UEs (based on Fig. 6.17 in [BD24]).

From a high level perspective, the MRT precoder aims at maximizing the signal power towards the spatial direction of each UE, which can result to non-negligible inter-UE interference. In contrast, the ZF precoder aims the signal power towards directions that are orthogonal to the spatial directions of the interfering UEs channels, thereby eliminating the interference, at the potential cost of noise amplification. The MMSE precoder offers a

tradeoff between the MRT and ZF approaches by incorporating noise variance knowledge in the precoder expression. Specifically, at high SNR it acts as the ZF precoder, while at low SNR it is equivalent to the MRT precoder.

As an example of the effectiveness of precoding methods in mitigating inter-UE interference, it is proposed to evaluate the effect of the ZF precoder. Taking the ZF precoder expression in Eq. (1.44) yields:

$$\begin{aligned}
\mathbf{y} &= \eta \mathbf{H}^{\mathsf{H}} \mathbf{W} \mathbf{x} + \mathbf{n} && (1.49)\\
&= \eta \mathbf{H}^{\mathsf{H}} \mathbf{H} \left(\mathbf{H}^{\mathsf{H}} \mathbf{H}\right)^{-1} \mathbf{x} + \mathbf{n} && (1.50)\\
&= \eta \mathbf{Id}_K \mathbf{x} + \mathbf{n} && (1.51)\\
&= \eta \mathbf{x} + \mathbf{n} && (1.52)\\
&\overset{(*)}{=} \frac{1}{\sqrt{\mathrm{Tr}\left((\mathbf{H}^{\mathsf{H}}\mathbf{H})^{-1}\right)}} \mathbf{x} + \mathbf{n}, && (1.53)
\end{aligned}$$

where $(*)$ originates from the normalization factor of the ZF precoder, as derived in Appendix A.4. One can observe that the ZF precoder effectively diagonalizes the channel, thereby fully removing inter-UE interference. However, this interference suppression comes at the cost of decreased received power when the user channels are correlated. Indeed, denoting the eigenvalues of $\mathbf{H}^{\mathsf{H}}\mathbf{H}$ by $\lambda_1, \ldots, \lambda_K$, channel correlation implies that some λ_i are small. Since the eigenvalues of $\left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1}$ are $\lambda_1^{-1}, \ldots, \lambda_K^{-1}$, and using the trace operator properties, the normalization factor becomes:

$$\eta = \left(\sqrt{\mathrm{Tr}\left((\mathbf{H}^{\mathsf{H}}\mathbf{H})^{-1}\right)}\right)^{-1} = \left(\sqrt{\sum_{i=1}^{K} \frac{1}{\lambda_i}}\right)^{-1}. \tag{1.54}$$

When several eigenvalues λ_i are small, the sum of their inverses becomes large, resulting in a low value of η. From a physical perspective, it means that the ZF precoder must amplify the power to compensate for the weak eigenmodes, effectively increasing the required transmit power. Under a fixed power budget, this amplification leads to reduced overall performance: as the useful signal $\mathbf{x}$ is scaled by η, a smaller η directly translates to a SNR loss.

The precoding schemes discussed herein assume full channel knowledge at the BS level, which necessitates a dedicated channel estimation procedure. This requirement introduces substantial complexity, particularly in FDD systems, where channel non-reciprocity further complicates estimation. This complexity constraint constitutes the problematic of Chapter 8 and 9. In addition to beamforming, inter-UE interference can also be mitigated through optimized transmit antenna array design. In particular, systems with high spatial resolution are better equipped to distinguish between closely spaced UEs, resulting in reduced inter-UE interference, even without the use of precoding techniques mitigating inter-UE interference. The concept of spatial resolution is presented in Highlight 1.7.

Highlight 1.7: Spatial resolution

A multi-antenna system is said to possess a high spatial resolution if its radiation diagram possess a narrow main lobe: the associated performance metric is the half-power beam-width (HPBW), i.e. the main radiation lobe aperture at -3dB. The HPBW depends on several factors:

1. The size of the antenna array that constitutes the multi-antenna system: as the array size increases, the main radiation lobe aperture decreases, leading to higher spatial resolution. For uniformly spaced antenna arrays, a larger number of radiating elements increases the array size and thus improves spatial resolution, as illustrated in Fig. 1.21.

2. The topology of the multi-antenna system: for instance uniform planar arrays have good spatial resolution in both the azimuth and elevation planes.

Note that, when considering beamforming towards an angle θ wrt. the normal of a planar antenna system, the HPBW broadens when $\theta \to \pm\pi/2$.

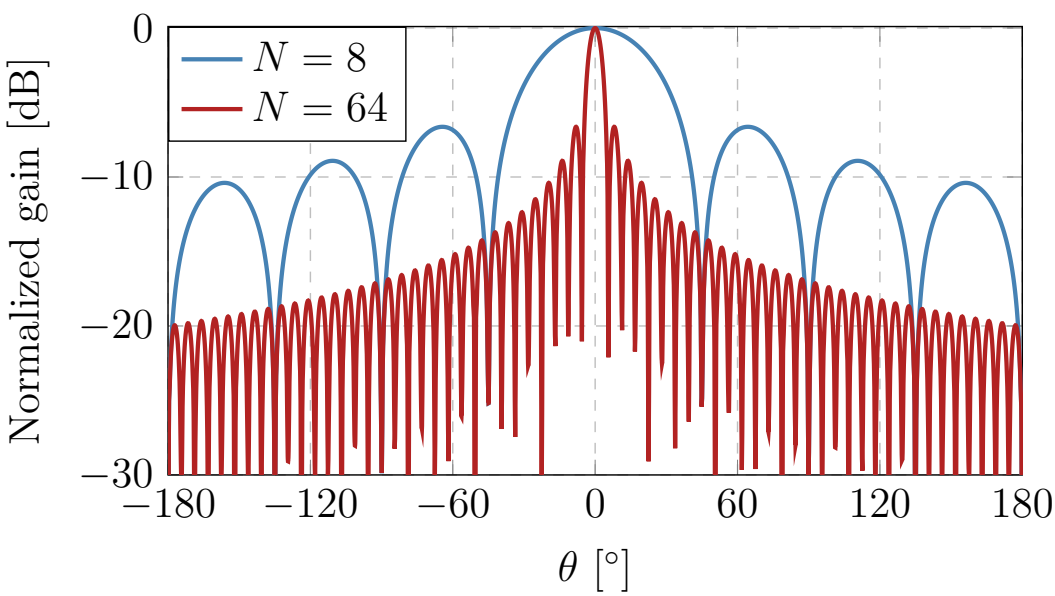


Figure 1.21: Illustration of the increase in spatial resolution with the size of the antenna array: in red, the system with 64 antennas presents a narrow main lobe, i.e. a low HPBW, and thus a high spatial resolution. Conversely, in blue, the system with 8 antennas presents a low spatial resolution due to its wide main lobe.

1.4 Conclusion

This introduction on the wireless propagation channel, and its use in modern communication systems, has highlighted several key aspects which are summarized below.

- The propagation channel is inherently linked to the physical environment, which can change over time, and is governed by fundamental physical phenomena such as reflection, absorption, refraction, diffraction, scattering, and the Doppler effect.
- It is possible to construct mathematical models rooted in these physical phenomena to represent the propagation channel, in both the time and frequency domains.
- It has been shown that many modern communication techniques can use channel knowledge to mitigate the limitations imposed by the propagation channel. Particularly, in MIMO systems, it was shown that water-filling or SVD beamforming allowed to optimize the capacity, while beamforming was shown to mitigate inter-UE interference in MU-MIMO systems. However these methods rely on accurate channel knowledge, which requires channel estimation procedures and induces a potentially significant overhead.

Chapter 2

Machine learning

This chapter aims at presenting a general introduction to the machine learning concepts. Firstly, Section 2.1 presents the different types of learning problems. Then, Section 2.2 introduces some general notions related to learning problems: e.g. how to train a machine learning method, how to select between different trained methods? Section 2.3 details three classical machine learning algorithms that are widely used for many problems. Section 2.4 focuses on the deep learning paradigm by providing a general presentation and stating its known limitations. Finally, Section 2.5 concludes on the presented concepts.

Contents

2.1 Typical problems **48**
- 2.1.1 Supervised learning 49
- 2.1.2 Unsupervised learning 51
- 2.1.3 Alternative learning paradigms 55

2.2 General learning notions **56**
- 2.2.1 Risk minimization 57
- 2.2.2 Learning method 61
- 2.2.3 Model selection criteria 63

2.3 Some popular ML methods **64**
- 2.3.1 Linear regression 64
- 2.3.2 k-nearest neighbors 65
- 2.3.3 Principal component analysis 67

2.4 Focus on deep learning **68**
- 2.4.1 Motivations 69
- 2.4.2 Popular architectures 75
- 2.4.3 Known limitations 76

2.5 Conclusion **78**

This chapter draws significant inspiration from the well-known and excellent book of Goodfellow, Bengio and Courville [GBC16], that constitutes a very good resource for machine learning basics and deep learning problematics. Additionally, the book of Shalev-Schwartz and Ben-David [SSBD14] is an excellent resource regarding machine learning theory. For french readers, the author recommends the very good book of Azencott [Aze19].

2.1 Typical problems

Over the past three years [1], there has been a remarkable surge in public interest in a class of machine learning (ML) algorithms known as *Generative AI*, largely driven by the emergence of powerful conversational agents such as ChatGPT, Claude, or LeChat. Despite this recent visibility, the field of machine learning is not new. In fact it can be traced back to the early 1960s with work in automated signal and speech recognition. However, scientific progress in the field has accelerated significantly over the past decade [2], an acceleration that can be attributed to several key factors. First, while previous generations of researchers were often constrained by limited computational resources, today's scientists benefit from highly powerful hardware, particularly due to the rapid advancements in graphical processing units (GPUs). These enhanced computational capabilities have significantly accelerated the development and deployment of complex machine learning models. Second, the range of practical problems that can be effectively addressed using ML techniques has expanded considerably. This has created a virtuous cycle: the problem-solving capabilities of ML methods have spurred greater interest in fundamental research, which in turn has produced more powerful and efficient algorithms, further enabling the resolution of real-world problems. An example of this cycle can be found in the field of computer vision: from a simple digit recognition system in [LeC+89], current ML-aided computer vision systems are used to operate self-driving cars. Third, as it will be discussed later in this chapter, ML methods are inherently data-driven. While data-collection has always been part of the human activity, the past fifty years have seen an unprecedented explosion in the volume of available data. This abundance of data has provided the basis for training increasingly expressive models, enabling ML methods to tackle more complex tasks with higher accuracy and robustness.

Before delving in the different types of machine learning problems, it is important to distinguish between artificial intelligence (AI) and ML. There are many different definitions of AI, but it can be considered as the field that aims to conceive intelligent systems, capable of performing complex tasks that requires traits typically linked with human intelligence: e.g. learning, reasoning, or decision taking. On the other hand, ML can be defined as a sub-field of AI, where the focus lies on developing methods that learn from data, in the goal of performing a task without specific instructions [3]. The core of ML can thus be split between four concepts [GBC16, Chapter 5, pp. 99-107]:

1. **Task.** A task constitutes the problem that will be solved through the use of a ML model.

2. **Data.** Data constitutes the elements that will be used to train a ML model.

1. At the time of writing this manuscript, the current year is 2025.
2. More precisely, since 2012 with the AlexNet paper [KSH12], which introduced the training of neural networks on GPUs.
3. Note that, while the training process may be guided by certain instructions or objectives, the execution of the task itself is typically carried out without explicit instructions.

3. **Learning algorithm.** A learning algorithm consists of an ordered sequence of operations that define the training procedure used to optimize a ML model for a specific task.

4. **Performance measure.** A performance measure quantifies how effectively a ML model performs the task for which it has been designed.

Machine learning problems can be categorized into different types according to the nature of the data and the underlying learning objective: see [GBC16, Chapter 5, pp. 104-107] and [Aze19, Chapter 1, pp. 4-10]. The main ML problem types are supervised learning, unsupervised learning, and alternative learning problems such as semi-supervised learning, reinforcement learning, or online learning. These problem types are presented in the subsequent subsections.

2.1.1 Supervised learning

In the supervised learning setup, a ML method is trained using *labeled* data: each input is associated to the corresponding target output, called a label. The objective is to learn a mapping from inputs to outputs, based on the available data, that enables accurate prediction on unseen data. The labels can be seen as an incentive towards a desired predictive behavior, and, as such, *supervise* the training process. A more formal definition of a supervised learning problem is proposed in Definition 2.1.

Definition 2.1: Supervised learning problem

Let $x \in \mathcal{X}$ be a variable from an input space $\mathcal{X}$, and $y \in \mathcal{Y}$ be a variable from an output space $\mathcal{Y}$. Let us assume that the relationship between x and y is captured by a joint distribution $p_{x,y}$ on $\mathcal{X} \times \mathcal{Y}$. The goal of supervised learning is to find a function f in a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ that minimizes a given error measure. Formally, this amounts to solving:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \mathbb{E}_{(x,y)\sim p_{x,y}} \left[\mu_{\mathcal{Y}} \left(f\left(x\right), y\right)\right], \\ \text{subject to} \quad & f \in \mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}} \end{aligned} \tag{2.1}$$

where $\mu_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ defines an error measure on $\mathcal{Y}$.

As presented in Definition 2.1, the approximation function f is chosen from an *hypothesis class* $\mathcal{H}$ which is a subset of mappings from $\mathcal{X}$ to $\mathcal{Y}$, i.e. $\mathcal{Y}^{\mathcal{X}}$. The choice of the hypothesis class is of fundamental importance as it constrains the set of functions that the learning algorithm can explore during training, thereby impacting the trained model's ability to efficiently approximate the unknown relation between $\mathcal{X}$ and $\mathcal{Y}$.

The nature of the input and output spaces typically depends on the task that is to be solved using supervised learning. Such tasks can be categorized in two main categories: regression and classification.

Regression

A regression task can be viewed as the prediction of a continuous output based on a given input, which translates to the following definition.

Definition 2.2: Regression problem

A supervised learning problem with output space $\mathcal{Y}$ is said to be a regression problem iff:

$$|\mathcal{Y}| = +\infty. \tag{2.2}$$

Some regression problems examples include:

- **Temperature prediction.** Predicting the average temperature of day d, based on temperature measured from the previous days $(d-1, d-2, \cdots, d-7)$, constitutes a regression problem with $\mathcal{X} = \mathbb{R}^7$, and $\mathcal{Y} = \mathbb{R}$.
- **Base station location optimization.** Given N base stations, the problem of optimizing their locations based on a performance measure $\kappa \in \mathbb{R}$ evaluated at K test users can be formulated as a regression problem with $\mathcal{X} = \mathbb{R}^K$, and $\mathcal{Y} = \mathbb{R}^{3\times N}$.
- **Constellation optimization** Selecting a set of N possible symbols, based on a measured performance $\kappa \in \mathbb{R}$ at a test UE, can be formulated as a regression problem with $\mathcal{X} = \mathbb{R}$, and $\mathcal{Y} = \mathbb{C}^N$.

Classification

A classification task can be interpreted as making a discrete decision based on a given input, as formalized in the following definition.

Definition 2.3: Classification problem

A supervised learning problem with output space $\mathcal{Y}$ is said to be a classification problem iff:

$$|\mathcal{Y}| < +\infty. \tag{2.3}$$

If $|\mathcal{Y}| = 2$ the problem is said to be a binary classification problem, whereas, if $2 < |\mathcal{Y}| < +\infty$, the problem is said to be a multi-class classification problem. Some examples of classification problem include:

- **Check fraud detection.** Determining whether or not a check is valid, based on an image of dimensions $N \times M$, is a binary classification problem [4] with $\mathcal{X} = \mathbb{R}^{3\times N\times M}$, and $\mathcal{Y} = \{0, 1\}$.

- **Tumor recognition.** Determining the type of a tumor based on a 3D image of dimensions $N \times M \times K$ obtained from magnetic resonance imaging constitutes a multi-class classification problem, where $\mathcal{X} = \mathbb{R}^{3\times N\times M\times K}$ and $\mathcal{Y} = \{0, 1, 2\}$ or $\mathcal{Y} = \{\text{benign}, \text{ malignant}, \text{ pre-cancerous}\}$.

- **Jammer recognition.** Determining if a received signal $\mathbf{x} \in \mathbb{C}^N$ originates from a jammer is a binary classification problem, with $\mathcal{X} = \mathbb{C}^N$, and $\mathcal{Y} = \{0, 1\}$.

2.1.2 Unsupervised learning

In the context of *unsupervised learning*, no labeled data are available, in contrast to the supervised learning framework. The objective is not to make predictions for a given input, but rather to identify hidden structures within the input space based solely on observations from the input space [GBC16, Chapter 5, pp. 146-147], [Aze19, Chapter 1, pp. 7-9]. A formal definition of this concept is proposed in Definition 2.4.

Definition 2.4: Unsupervised learning

Let $x \in \mathcal{X}$ be a variable from an input space $\mathcal{X}$. Let us assume that the data distribution is captured by an unknown probability distribution p_x on $\mathcal{X}$, and that there exists a representation space $\mathcal{Z}$, in which relevant information about p_x can be captured. The goal of unsupervised learning is to identify a function f, within an hypothesis class $\mathcal{H} \subseteq \mathcal{Z}^{\mathcal{X}}$, such that f minimizes a task-specific objective function. Formally, this amounts to solving:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \mathbb{E}_{x\sim p_x}\left[\mu_{\mathcal{Z}}\left(f\left(x\right)\right)\right], \\ \text{subject to} \quad & f \in \mathcal{H} \subseteq \mathcal{Z}^{\mathcal{X}} \end{aligned} \tag{2.4}$$

where $\mu_{\mathcal{Z}} : \mathcal{Z} \to \mathbb{R}^+$ quantifies how well the learned representation captures the underlying structure of the data.

4. Note that, in this example, each pixel is considered to be encoded as a real scalar. In practice it is typically encoded on 8 bits, thereby limiting the input space to $\{0, \cdots, 255\}^{3\times N\times M}$.

Note that the concept of unsupervised learning, and the notion of mapping function to a representation space are quite broad and abstract: it encompasses not only analytically defined functions, but also those that do not admit a closed-form expression. Examples of unsupervised learning problems include density estimation [Ros56; Par62], clustering [Mac67], dimensionality reduction [Pea01], data denoising [CLMR23], or sparse representation learning. A quick focus on density estimation, clustering, and dimensionality reduction is proposed in the following paragraphs.

Density estimation

Assuming that the input space $\mathcal{X}$ is governed by an unknown probability density function $p_x(x)$, i.e.:

$$\forall x \in \mathcal{X}, x \sim p_x(x), \tag{2.5}$$

and given an observation of the input space, $\mathcal{D} = \{x_1, \cdots, x_N\} \in \mathcal{X}^N$, the density estimation task consists in estimating a surrogate density function $f(x)$ using $\mathcal{D}$. In this context, the representation space $\mathcal{Z}$ is a latent space in which the data distribution $p_x(x)$ is easier to model. From a more formal perspective, the density estimation task can be seen as solving the following optimization problem:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mu_{\mathcal{Z}}(f(x)), \\ \text{subject to} \quad & f \in \mathcal{H} \end{aligned} \tag{2.6}$$

where $\mathcal{H}$ is the hypothesis class associated to this problem [5], and $\mu_{\mathcal{Z}} : \mathcal{Z} \to \mathbb{R}^+$ denotes an error measure, e.g. the Kullback-Leibler divergence with the empirical probability density function evaluated on $\mathcal{D}$. The objective is that the solution of the optimization problem presented in Eq. (2.6), $f^\star$, approximates the true data density function in the sense that the following Kullback-Leibler divergence tends to 0:

$$\int_{\mathcal{X}} f^\star(x) \ln\left(\frac{f^\star(x)}{p_x(x)}\right) \mathrm{d}x \simeq 0. \tag{2.7}$$

A classical example of density estimation method is the kernel density estimator, introduced independently by Rosenblatt and Parzen [Ros56; Par62], and later extended to the multivariate setting [Sco15]. An illustration of this task is proposed in Fig. 2.1.

Clustering

The clustering task seeks to group data samples according to shared characteristics. A general illustration of this process is presented in Fig. 2.2. Formally, clustering involves finding a function $f\colon \mathcal{X} \to \mathcal{Z} = [\mathcal{C}_1, \cdots, \mathcal{C}_K]$, which assigns each data point $x \in \mathcal{X}$ to one

5. Note that in the context of this problem, $\mathcal{H}$ can be defined as the set of all probability density functions, i.e. $\mathcal{H} = \left\{f \in \mathcal{Z}^{\mathcal{X}} | \forall x \in \mathcal{X}, f(x) \geq 0, \int_{\mathcal{X}} f(x)\,\mathrm{d}x = 1\right\}$.

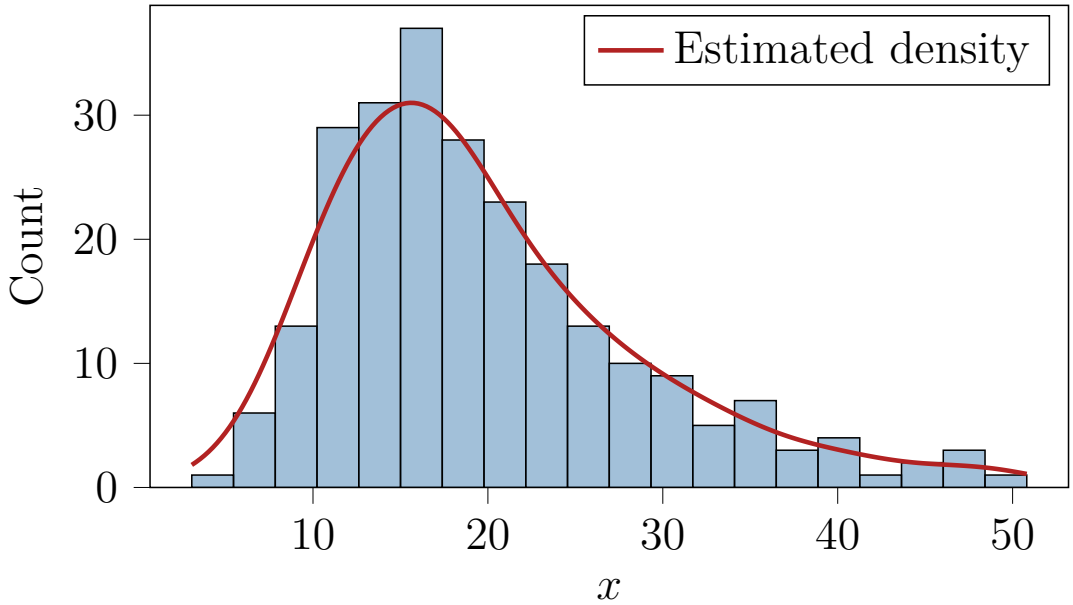


Figure 2.1: Illustration of the density estimation task: the histogram is based on the observation $\mathcal{D}$, and the red curve represents the estimated density f (based on [YuHsu25]).

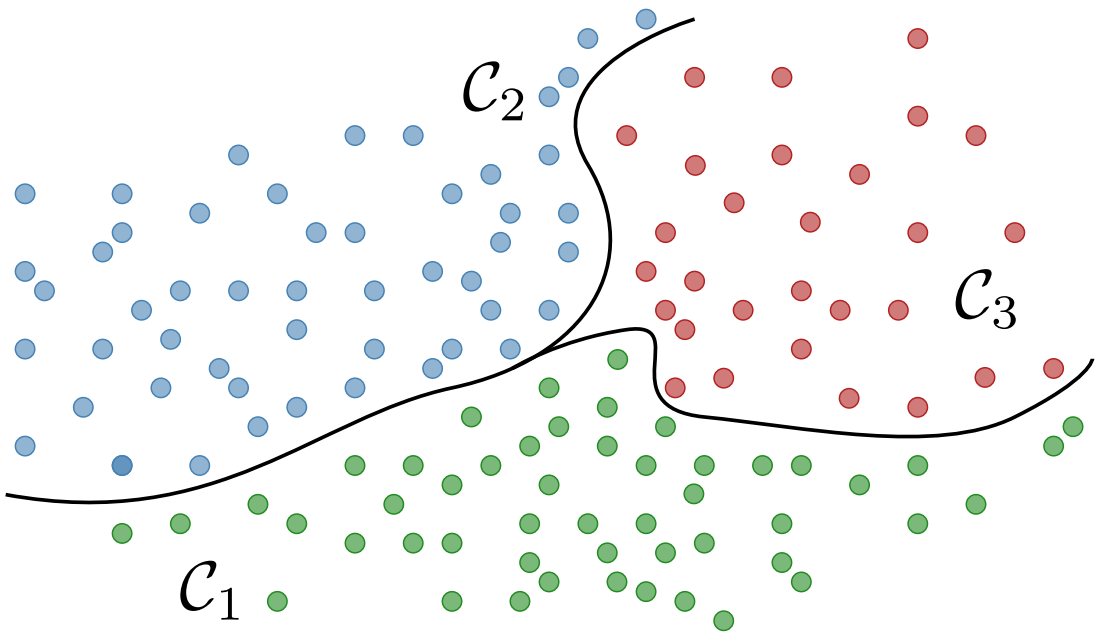


Figure 2.2: Illustration of the clustering task: given an observation $\mathcal{D}$ (here points in $\mathbb{R}^2$), the objective is to separate them in different clusters $\mathcal{C}_k$, such that each point in the same cluster shares a common characteristic (here its color).

of the K clusters $\mathcal{C}_k$, based on a certain similarity criterion. The objective is to identify a partition of the input space from an observation $\mathcal{D} = \{x_1, \cdots, x_N\} \in \mathcal{X}^N$, such that:

$$\bigcup_{k=1}^{K} \mathcal{C}_k \simeq \mathcal{X}, \tag{2.8}$$

and:

$$\mathcal{C}_i \cap \mathcal{C}_j = \varnothing, \forall i \neq j. \tag{2.9}$$

Formally, the clustering task takes the form of the following optimization problem:

$$\begin{aligned} \underset{f,\,(\mathcal{C}_1,\cdots,\mathcal{C}_K)}{\text{minimize}} \quad & \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mu_{\mathcal{Z},\mathcal{X}^K} \left(f\left(x\right), \left(\mathcal{C}_1, \cdots, \mathcal{C}_K\right)\right), \\ \text{subject to} \quad & f \in \mathcal{H}, \\ & \left(\mathcal{C}_1, \cdots, \mathcal{C}_K\right) \subset \mathcal{X}^K, \\ & \bigcup_{k=1}^{K} \mathcal{C}_k \simeq \mathcal{X}, \\ & \mathcal{C}_i \cap \mathcal{C}_j = \varnothing, \forall i \neq j \end{aligned} \tag{2.10}$$

where $\mathcal{H}$ denotes the hypothesis class, and $\mu_{\mathcal{Z},\mathcal{X}^K} \colon \mathcal{Z} \times \mathcal{X}^K \to \mathbb{R}^+$ denotes an error measure on the cluster assignment. The influence of the aforementioned similarity criterion manifests through the design of this error measure, which determines how similarity or dissimilarity between samples is evaluated during the optimization process.

Classical clustering methods include the k-means method [Mac67], or the gaussian mixture model method [Ras99]. Examples of clustering applications include:

- **Image segmentation.** Given an image, separate its semantic content in different regions, e.g. road, pedestrian, or stop sign.

- **Customer segmentation.** Given a purchase records, separate customers within distinct regions to give business analytics such as product preferences per buyer profile.

Dimensionality reduction

The general objective of the dimensionality reduction task is to find a representation space whose dimension is lower than the input space, and where some properties of the input space are preserved: an illustration of this task is presented in Fig. 2.3. Given an observation $\mathcal{D} = \{x_1, \cdots, x_N\} \in \mathcal{X}^N$, the dimensionality reduction task can be formally defined as follows:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mu_{\mathcal{Z},\mathcal{X}} \left(f\left(x\right), x\right), \\ \text{subject to} \quad & f \in \mathcal{H} = \left\{ f \in \mathcal{Z}^{\mathcal{X}} \middle| \dim_{\mathbb{R}}\left(\mathcal{Z}\right) < \dim_{\mathbb{R}}\left(\mathcal{X}\right) \right\} \end{aligned} \tag{2.11}$$

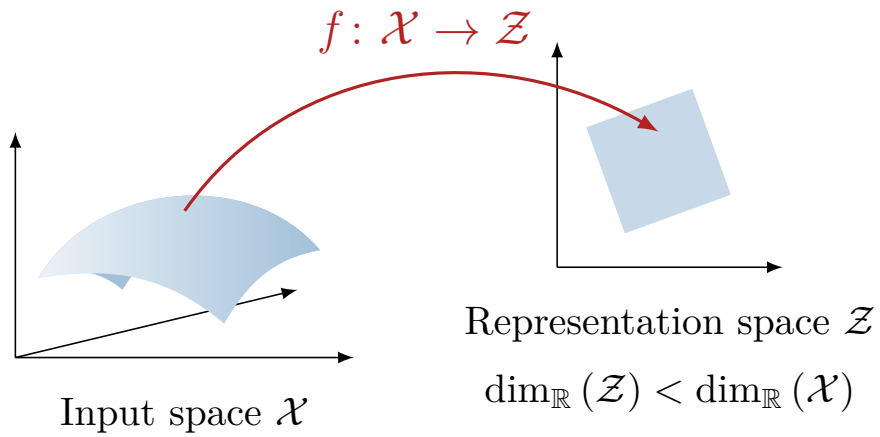


Figure 2.3: Illustration of the dimensionality reduction task (based on [Fle25]): the observation $\mathcal{D}$ from the input space (here $\mathcal{X} = \mathbb{R}^3$), is mapped towards a lower dimensional space (here $\mathcal{Z} = \mathbb{R}^2$).

where $\mu_{\mathcal{Z},\mathcal{X}} : \mathcal{Z} \times \mathcal{X} \to \mathbb{R}^+$ denotes an error measure on the *encoded* input. In a similar manner than for the clustering task, the design of this error measure is highly dependent on the specific objective of the dimensionality reduction task. Different goals, such as preserving pairwise distances, capturing variance, or maintaining neighborhood structure, may require distinct formulations of $\mu_{\mathcal{Z},\mathcal{X}}$. Applications of dimensionality reduction include:

- **Compression tasks.** Storing the encoded representation requires less memory than storing the original data. In this context, the encoding function must admit an inverse so that it is possible to decode the encoded information. Additionally, the decoded data must present acceptable distortion [6].
- **Channel charting.** This method aims at embedding high dimensional channel matrices on low dimensional representation spaces, with topology preservation constraints. It is the core concept of the contributions presented in Chapters 8 and 9.

2.1.3 Alternative learning paradigms

Semi-supervised learning

Hitherto, learning problems have been categorized according to the availability of labels. Nevertheless, mixed scenarios may arise, in which only a subset of the data is labeled while the remainder remains unlabeled. This setting constitutes the foundation of semi-supervised learning: see [CSZ06] for more information. Semi-supervised learning is of

6. Note that, besides its dimensionality, the entropy of the encoded variable must also be taken into account when optimizing the compression scheme.

particular interest in some problems where data collection is easy, but annotation is tedious: e.g. computer vision problems, where large image or video datasets are available, but where data annotation is a very time-consuming process.

Reinforcement learning

Another important learning paradigm is reinforcement learning. This framework is quite similar to a human learning process: an *agent* interacts with an *environment* by selecting *actions* in a fixed set. After one or multiple actions, the agent receives a *reward* that can be either positive or negative. The design of the reward policy constitutes the core this learning paradigm, as it guides the agent towards a desired behavior. An interested reader can refer to [SB18; BT96] for more information [7]. For instance, this paradigm is used to train modern Go/chess/shogi engines: the environment is the Go/chess/shogi game [Sil+16; Sil+18], the actions are the game's possible plays regarding the current agent's situation, and the reward constitutes a feedback regarding the agent's actions.

Online learning

Online learning constitutes a learning paradigm that can be formulated in both supervised and unsupervised settings. Its central principle is to avoid relying on a fixed dataset $\mathcal{D}$, and instead updating the model continuously as new data becomes available. This approach is particularly well-suited for systems where data arrives sequentially over time. For instance, one can consider a transmission quality predictor deployed on a base station. In the classical supervised or unsupervised settings, training such a predictor requires the prior collection of a large dataset of received signals, which may be time-consuming. By contrast, within the online learning paradigm, the predictor's parameters are updated incrementally with each newly received signal, enabling real-time adaptation to changing conditions.

2.2 General learning notions

Now that notions regarding the different learning paradigms have been presented, it is proposed to present some elements about the actual learning process. Note that this section will focus on supervised learning problems, but that the same notions can be extended in the other frameworks.

7. Note that reinforcement learning can be seen as a generalization of the Markov decision process framework, when the transition probabilities of the environment are unknown, or when the state space is too large to be exhaustively explored.

2.2.1 Risk minimization

Towards the empirical risk

As presented in Definition 2.1, solving a supervised learning task amounts to finding a mapping $f\colon \mathcal{X} \to \mathcal{Y}$, that minimizes the expected error measure over the data distribution $p_{x,y}$. This expected error is referred to as the *risk*, and quantifies how well the learned mapping captures the underlying data relations between $\mathcal{X}$ and $\mathcal{Y}$.

Definition 2.5: Risk

Let $p_{x,y}$, defined over $\mathcal{X} \times \mathcal{Y}$, be the data distribution in a supervised learning problem, where $f \in \mathcal{Y}^{\mathcal{X}}$ is the learned mapping, $\mathcal{X}$ being the input space, and $\mathcal{Y}$ being the output space. Given an error measure $\mu_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$, the risk is defined as:

$$\mathsf{R}\left(f\right) = \mathbb{E}_{(x,y)\sim p_{x,y}}\left[\mu_{\mathcal{Y}}\left(f\left(x\right), y\right)\right]. \tag{2.12}$$

Remark 2.1. *The error measure is also commonly referred to as the loss function, denoted by $\mathcal{L}$. To avoid confusion with the set notation, and to maintain consistency with the error measure presented in Section 2.1, the $\mu_{\mathcal{Y}}$ notation is used thereafter, but the error measure and loss function terms are interchangeably used.*

The key difference between optimization and machine learning lies in Definition 2.5. If the true data distribution $p_{x,y}$ is known, minimizing the risk over the hypothesis class $\mathcal{H}$ constitutes a pure optimization problem. However, in most ML problems, one only has access to $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \in \mathcal{X}^N \times \mathcal{Y}^N$, consisting of N i.i.d. samples drawn from $p_{x,y}$. In this case, the true data distribution is unavailable, and one instead works with an empirical estimate $\hat{p}_{x,y|\mathcal{D}}$, which depends on the observation $\mathcal{D}$ [8]. The learning problem therefore translates to the following *empirical risk minimization* problem:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \mathbb{E}_{(x,y)\sim \hat{p}_{x,y|\mathcal{D}}}\left[\mu_{\mathcal{Y}}\left(f\left(x\right), y\right)\right] = \frac{1}{|\mathcal{D}|}\sum_{(x,y)\in\mathcal{D}} \mu_{\mathcal{Y}}\left(f\left(x\right), y\right). \\ \text{subject to} \quad & f \in \mathcal{H} \end{aligned} \tag{2.13}$$

Remark 2.2. *It can be observed that the optimization problem presented in Eq.* (2.13) *is similar to those presented in Eq.* (2.6), (2.10), *and* (2.11). *Indeed, all these problems can be formulated as instances of empirical risk minimization.*

The central idea behind empirical risk minimization is to consider that the observation $\mathcal{D}$ successfully captures the underlying true data distribution. Under this assumption,

8. The index $\mathcal{D}$ in $\hat{p}_{x,y|\mathcal{D}}$ denotes the distribution estimate dependence to the observation $\mathcal{D}$.

minimizing the empirical risk serves as a surrogate for minimizing the true risk. Nonetheless, several important considerations must be highlighted regarding the empirical risk minimization principle:

1. With a sufficiently large number of observations, i.e. $|\mathcal{D}| \to +\infty$, the empirical risk converges to the risk: this follows directly from the strong law of large numbers.

2. This asymptotic convergence generally does not extend to the minimizers of the respective risk functions. In particular, a minimizer of the empirical risk is not necessarily statistically consistent, i.e. it does not converge in probability to a minimizer of the true risk as $|\mathcal{D}| \to +\infty$. Notwithstanding, given certain assumptions on the hypothesis class $\mathcal{H}$, theoretical convergence guarantees can be established: an interested reader is referred to the Vapnik-Chervonenkis theory [Vap00].

3. By definition, the empirical risk minimization is ill-posed: multiple minimizers of the empirical risk may exist, leading to the formulation of Highlight 2.1.

Highlight 2.1: Empirical risk minimization and machine learning

In machine learning problems, the goal is to minimize the empirical risk with the hope that this will also lead to the minimization of the true risk.

Generalization and over/underfitting

As stated before, in supervised ML methods, the objective is to train a model, based on observations $\mathcal{D}$, to predict a behavior on unseen data. For a trained model, the quality of its predictions on unseen data constitutes its *generalization* performance.

Definition 2.6: Generalization error

Let $f \in \mathcal{Y}^{\mathcal{X}}$ be a trained ML model using the observation $\mathcal{D} \in \mathcal{X}^N \times \mathcal{Y}^N$. Given a certain generalization performance measure $\mu_{\mathcal{Y}}^{\mathrm{g}} \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$, the generalization error on unseen data $\mathcal{G} \in \mathcal{X}^M \times \mathcal{Y}^M$ is defined as:

$$\epsilon_{\mathrm{g}}\left(f|\mathcal{G}\right) = \frac{1}{|\mathcal{G}|} \sum_{(x,y)\in\mathcal{G}} \mu_{\mathcal{Y}}^{\mathrm{g}}\left(f\left(x\right), y\right). \tag{2.14}$$

The generalization notion gives rise to the concepts of *overfitting* and *underfitting*. A model that achieves a very good empirical risk value [9], but a high generalization error is

9. This value is commonly referred to as the training error.

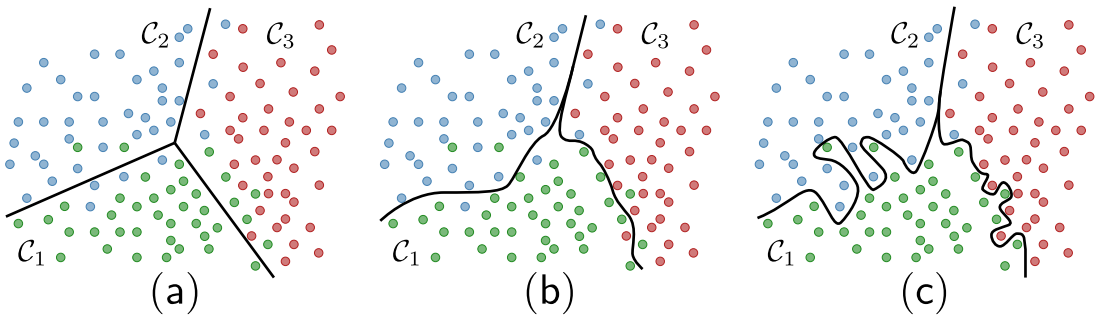


Figure 2.4: Illustration of the overfitting/underfitting concept on a clustering task: (a) underfitting, (b) optimal fit, (c) overfitting.

said to overfit the training data. In such cases, the trained model performs well on the data it was trained on, but fails to predict on unseen examples, thereby limiting its actual utility. Conversely, a model that fails to achieve a low training error is said to underfit the training data. Such cases usually happen when the model's complexity is insufficient relative to that of the actual mapping: by design, the model is not able to capture the underlying data structures, resulting in underfitting. This overfitting/underfitting concept is illustrated in Fig. 2.4.

Highlight 2.2: Generalization, overfitting, and underfitting

Given a certain ML model:

- Generalization refers to the model's ability to perform well on unseen data, beyond the training set.
- Overfitting refers to the case where the trained model is able to learn on the training data, but exhibits poor generalization performance.
- Underfitting refers to the case where the model is too simplistic to capture the underlying structure of the training data, leading to poor performance even on the training set.

Bias-variance tradeoff

A well known principle in ML theory is the concept of bias-variance trade-off [10]: this tradeoff can be defined and explained from the *excess risk* perspective. Given a trained model $f \in \mathcal{Y}^{\mathcal{X}}$, the excess risk is defined as the difference between the optimal risk, i.e. risk evaluated at the function that minimizes the risk over the set of all functions:

10. See [SSBD14, Chapter 5, pp. 64-65] for more information.

$\mathsf{R}^\star = \inf_{f \in \mathcal{Y}^\mathcal{X}} \mathsf{R}(f)$, and the risk evaluated on f, i.e. $\mathsf{R}(f)$. The excess risk admits an approximation/estimation error decomposition, formalized in Definition 2.7.

Definition 2.7: Excess risk decomposition

Let $f \in \mathcal{Y}^\mathcal{X}$ be a trained ML model, belonging to an hypothesis class $\mathcal{H}$, $\mathsf{R}(f)$ being its associated risk. Let $\mathsf{R}^\star$ be the optimal risk, i.e.:

$$\mathsf{R}^\star = \inf_{f \in \mathcal{Y}^\mathcal{X}} \mathsf{R}(f). \tag{2.15}$$

The excess risk is defined as $\mathsf{R}(f) - \mathsf{R}^\star$, and decomposes as:

$$\mathsf{R}(f) - \mathsf{R}^\star = \left(\mathsf{R}(f) - \inf_{f \in \mathcal{H}} \mathsf{R}(f)\right) + \left(\inf_{f \in \mathcal{H}} \mathsf{R}(f) - \mathsf{R}^\star\right) \tag{2.16}$$

$$= \epsilon_{\text{est}}(f) + \epsilon_{\text{app}}(f), \tag{2.17}$$

where $\epsilon_{\text{est}}(f)$ denotes an estimation error, and $\epsilon_{\text{app}}(f)$ an approximation error.

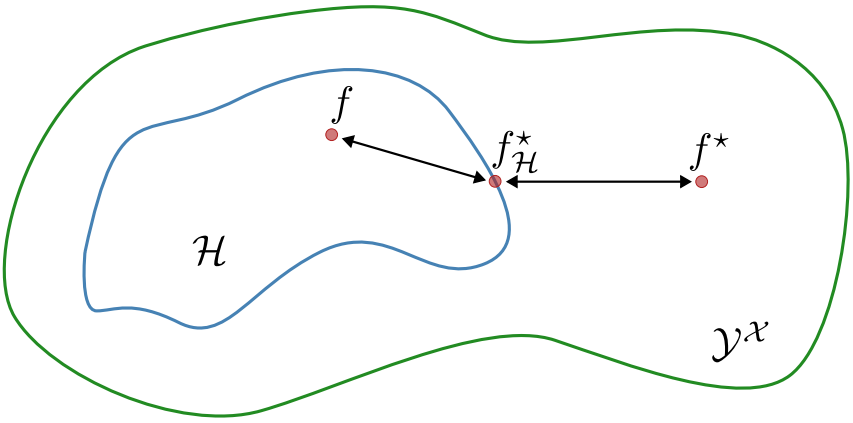


Figure 2.5: Illustration of Definition 2.7: f is the considered model, $f^\star_\mathcal{H} = \arg\inf_{f \in \mathcal{H}} \mathsf{R}(f)$ represents the optimal model (from a risk perspective) within the hypothesis class $\mathcal{H}$, and $f^\star = \arg\inf_{f \in \mathcal{Y}^\mathcal{X}} \mathsf{R}(f)$ is the optimal model (from a risk perspective) within $\mathcal{Y}^\mathcal{X}$ (based on [Rak12]).

Definition 2.7 is illustrated in Fig. 2.5, and several comments can be made:

- The estimation error, $\epsilon_{\text{est}}(f) = \mathsf{R}(f) - \inf_{f \in \mathcal{H}} \mathsf{R}(f)$, quantifies the distance between the trained model and the optimal model in the hypothesis class. Considering a large observation size, i.e. $|\mathcal{D}| \to +\infty$, tends to reduce this error.

- The approximation error, $\epsilon_{\text{app}}(f) = \inf_{f \in \mathcal{H}} \mathsf{R}(f) - \mathsf{R}^\star$, quantifies the distance between the optimal model in the hypothesis class and the optimal model over the set of all possible functions, i.e. it describes the quality of the hypothesis class. If $\mathcal{H} = \mathcal{Y}^{\mathcal{X}}$, then the approximation error vanishes.

The excess risk can thus be decomposed into two components: a term that measures the effectiveness of the learning method, i.e. how well the learning method is able to find a solution close to the optimal one within the hypothesis class, and another term that evaluates expressiveness of the hypothesis class, i.e. how well the hypothesis class is adapted to the considered problem. The well known bias-variance problem then arises: selecting a large hypothesis class reduces the approximation error (low bias), but generally increases the estimation error (high variance) as the support of the search space is higher. Conversely, selecting a small hypothesis class yields a low estimation error (low variance) due to a small search space support, but increases the approximation error (high bias) due to the limited representation capabilities of the optimal function in the hypothesis class.

2.2.2 Learning method

Solving the problem represented in Eq. (2.13) constitutes the objective of the *learning method*: given the observation $\mathcal{D}$ and the hypothesis class $\mathcal{H}$, the goal is to minimize the empirical risk. Depending on the loss function and hypothesis class properties, the problem may admit a closed form analytical solution: e.g. this is the case for the linear regression problem, presented in details in Section 2.3.1. However, most modern ML models are too complex to allow for such analytical solutions. In these cases, gradient-based optimization methods are usually used to iteratively approximate a solution. Indeed, given that the hypothesis class consists of parametric functions [11], where $\boldsymbol{\theta}$ denotes the learnable parameters, the optimization problem rewrites as:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{minimize}} \quad & \mathbb{E}_{(x,y)\sim\hat{p}_{x,y|\mathcal{D}}}\left[\mu_{\mathcal{Y}}\left(f_{\boldsymbol{\theta}}(x), y\right)\right], \\ \text{subject to} \quad & \boldsymbol{\theta} \in \mathcal{H}_\theta \end{aligned} \tag{2.18}$$

where $\mathcal{H}_\theta$ represents the hypothesis class parameter space, i.e. $\mathcal{H} = \{f_{\boldsymbol{\theta}} | \boldsymbol{\theta} \in \mathcal{H}_{\boldsymbol{\theta}}\}$. Under the hypothesis of error measure convexity wrt. $\boldsymbol{\theta}$, an approximated solution of Eq. (2.18) can be obtained by finding $\boldsymbol{\theta}$ for which the empirical risk gradient vanishes, i.e. solving $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{(x,y)\sim\hat{p}_{x,y|\mathcal{D}}}\left[\mu_{\mathcal{Y}}\left(f_{\boldsymbol{\theta}}(x), y\right)\right] = \mathbf{0}$ for $\boldsymbol{\theta}$.

Remark 2.3. *Note that gradient-based methods are also used when the error measure is non-convex, with the hope of converging towards a satisfying local minimum. This constitutes the core of deep learning methods, presented in details in Section 2.4.*

11. Note that such assumption is weak as many ML methods consists of parametric models.

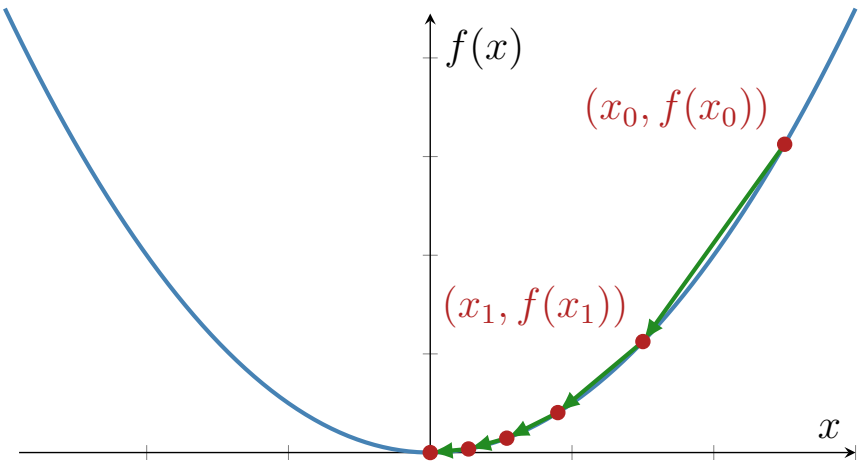


Figure 2.6: Illustration of the gradient descent algorithm: x_0 is the initialization point, and x_i represents the obtained location after i gradient updates.

Algorithm 1 Gradient descent.

Inputs: Function to minimize f, initial value $x \in \mathcal{X}$, stopping criterion `SC`

```
while SC = False do
    Compute the gradient update: ∇_x f(x)
    Choose an update step: λ
    Update the current value: x ← x − λ∇_x f(x)
end while
```

Output: x (Approximated minimum).

Gradient descent algorithm

Due to its extensive use in machine learning, it is proposed to quickly present the gradient descent algorithm. Given a convex function $f \in \mathcal{Y}^{\mathcal{X}}$, an approximated solution to the following optimization problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x), \\ \text{subject to} \quad & x \in \mathcal{X} \end{aligned} \tag{2.19}$$

can be obtain using Algorithm 1. An illustration of the gradient descent concept is proposed in Fig. 2.6. The choice of the update step [12] $\lambda \in \mathbb{R}^+$ is of paramount importance: a value that is too large may lead to divergence or unstable behavior, while a value that is too small can result in excessively slow convergence. Note that modern gradient descent algorithms, such as Adam [KB15], include mechanisms to adapt the update step dynamically during training, often leading to faster and more stable convergence.

12. Also commonly referred to as the learning rate.

2.2.3 Model selection criteria

Previous sections have addressed the definition of learning problems, key theoretical notions regarding learning, and of the learning process itself. The focus now shifts towards the question of model selection, that is, how to choose among several trained models. Indeed, the famous *no free lunch* theorem [WM97] [13], states that for every ML method, there exists a task on which it fails: i.e. no model can have good performance on all tasks. It thus important to define selection methods between several models.

Training, test and validation sets

Given a supervised learning problem with hypothesis class $\mathcal{H}$, the aim is to find a mapping f that minimizes the empirical risk over the observation $\mathcal{D}$, also denoted as the training set, but that also presents a low generalization error. Evaluating the generalization error requires data not seen during the training phase: this motivates the need for a test set, independent from the training set. Now, let us assume that K models have been selected from $\mathcal{H}$, trained on the training set, and whose generalization errors have been evaluated on the test set. The central challenge lies in selecting the most appropriate model between the K trained candidates. A solution consists in defining a third set, known as the validation set, independent from the training and test sets, and used to compute an evaluation measure.

Definition 2.8: Training, test, and validation sets

Let $\mathcal{D} \in \mathcal{X}^N \times \mathcal{Y}^N$ be the available data for a supervised learning problem. The training, test, and validation sets, $\mathcal{D}_{\text{tr}}$, $\mathcal{D}_{\text{te}}$ and $\mathcal{D}_{\text{val}}$, are defined by partitioning $\mathcal{D}$ such that:

$$\mathcal{D}_{\text{tr}} \cap \mathcal{D}_{\text{te}} = \varnothing \tag{2.20}$$

$$\mathcal{D}_{\text{tr}} \cap \mathcal{D}_{\text{val}} = \varnothing \tag{2.21}$$

$$\mathcal{D}_{\text{te}} \cap \mathcal{D}_{\text{val}} = \varnothing. \tag{2.22}$$

Given K parametric models: $f_{\boldsymbol{\theta}_k} \in \mathcal{H}_k \subset \mathcal{H}, k \in [\![1, K]\!]$, a simple model selection method, known as the *hold-out* method, consists of the following steps:

1. Train each model on the training set, using the error measure $\mu_{\mathcal{Y}}$:

$$\forall k \in [\![1, K]\!], f^{\star}_{\boldsymbol{\theta}_k} = \underset{f_{\boldsymbol{\theta}_k} \in \mathcal{H}_k}{\arg\min} \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{(x,y) \in \mathcal{D}_{\text{tr}}} \mu_{\mathcal{Y}} \left(f_{\boldsymbol{\theta}_k}(x), y \right). \tag{2.23}$$

13. An interested reader is referred to [SSBD14, Chapter 5, pp. 61-63] for a formal definition.

2. Select the model that has the lowest error on the validation set:

$$f^{\star}_{\boldsymbol{\theta}_{k^\star}} = \underset{f^{\star}_{\boldsymbol{\theta}_k} \in \left\{ f^{\star}_{\boldsymbol{\theta}_1}, \cdots, f^{\star}_{\boldsymbol{\theta}_K} \right\}}{\arg\min} \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(x,y) \in \mathcal{D}_{\text{val}}} \mu_{\mathcal{Y}} \left(f^{\star}_{\boldsymbol{\theta}_k}(x), y \right). \tag{2.24}$$

3. Evaluate the generalization error of the selected model on the test set:

$$\epsilon_{\text{g}} \left(f^{\star}_{\boldsymbol{\theta}_{k^\star}} | \mathcal{D}_{\text{te}} \right) = \frac{1}{|\mathcal{D}_{\text{te}}|} \sum_{(x,y) \in \mathcal{D}_{\text{te}}} \mu^{\text{g}}_{\mathcal{Y}} \left(f^{\star}_{\boldsymbol{\theta}_{k^\star}}(x), y \right). \tag{2.25}$$

Note that more sophisticated methods, e.g. k-fold cross-validation, exist in order to reduce the amount of required data for the validation set. An interested reader can refer to [GBC16, Chapter 5, pp. 122-123] or [SSBD14, Chapter 11, pp. 149-150] for more information.

Complexity regularization

During the model selection step, each model complexity can be taken into account, following the Occam's razor principle: "*Entities are not to be multiplied without necessity*", i.e. given two models with similar performance, the least complex is to be preferred. Mathematically, this can be accomplished by adding a complexity regularization term in the score function used during the model selection step, i.e. $\mu_{\mathcal{Y}}$ in Eq. (2.24): popular complexity regularizers include the Akaike information criterion or the Bayesian information criterion. An interest reader is referred to [SS04] for more information.

2.3 Some popular ML methods

After laying the theoretical foundations of machine learning, this section proposes illustrative examples with three well-known ML methods: linear regression, k-nearest neighbors (k-NN), and principal component analysis (PCA).

2.3.1 Linear regression

Linear regression represents the simplest from of regression problems, with origins tracing back to the early 19th century. It is a non-parametric supervised learning problem that amounts to modeling an output variable $\mathbf{y} \in \mathbb{R}^m$ as a linear combination of an input vector $\mathbf{x} \in \mathbb{R}^l$. Given an observation $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N \in \mathbb{R}^{l \times N} \times \mathbb{R}^{m \times N}$, this learning problem formally reads as:

$$\begin{aligned} \underset{\mathbf{W}}{\text{minimize}} \quad & \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} \|\mathbf{y} - \mathbf{W}\mathbf{x}\|_2^2, \\ \text{subject to} \quad & \mathbf{W} \in \mathbb{R}^{m \times l} \end{aligned} \tag{2.26}$$

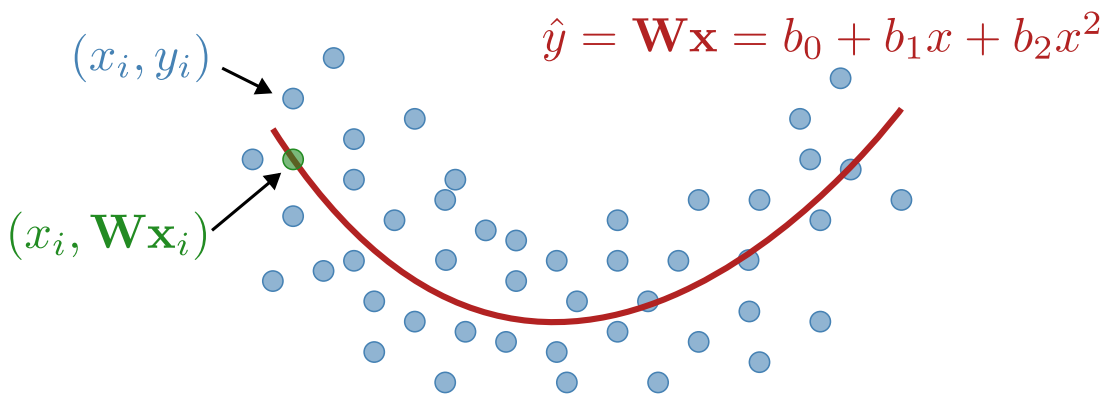


Figure 2.7: Illustration of the regression concept. The goal is to learn a representation of the relation between $x_i \in \mathbb{R}$ and $y_i \in \mathbb{R}$: the input space is $\mathcal{X} = \mathbb{R}^3$, such that $\mathbf{x}_i = (1, x_i, x_i^2)$, and the output space is $\mathcal{Y} = \mathbb{R}$, with $\mathbf{W} = [b_0, b_1, b_2]$, such that $\hat{y}_i = b_0 + b_1 x_i + b_2 x_i^2$.

where $\mathbf{W} \in \mathbb{R}^{m \times l}$ is a weight matrix. In this problem, the input space is $\mathcal{X} = \mathbb{R}^l$, the output space is $\mathcal{Y} = \mathbb{R}^m$, and the loss function is the squared ℓ_2 norm. Solving this problem is straightforward as it admits the following analytical solution:

$$\mathbf{W}^\star = \mathbf{Y}\mathbf{X}^\mathsf{T} \left(\mathbf{X}\mathbf{X}^\mathsf{T}\right)^{-1}, \tag{2.27}$$

where $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N] \in \mathbb{R}^{m \times N}$, and $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{l \times N}$. A demonstration of this solution is provided in Appendix B.1, and an illustration is proposed in Fig. 2.7.

2.3.2 k-nearest neighbors

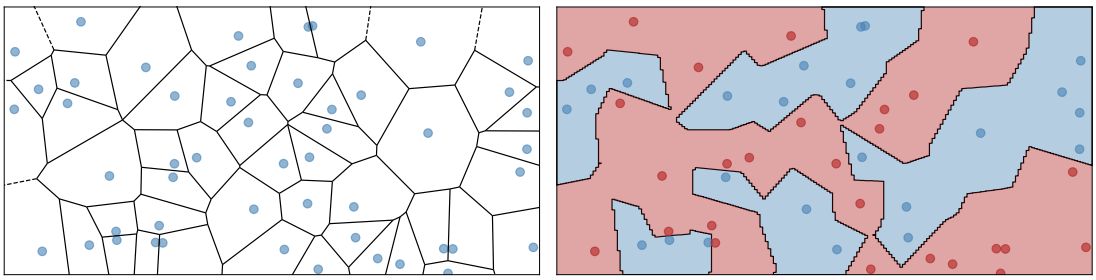
k-NN is a very simple, yet powerful, supervised ML method that can solve both classification and regression problems. Given an observation $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \in \mathcal{X}^N \times \mathcal{Y}^N$, the objective of k-NN is to associate a label $y \in \mathcal{Y}$ to an input $x \in \mathcal{X}$ based on the labels of the k closest locations $x_k \in \mathcal{D}$ with respect to a distance [14] $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$.

Nearest neighbor concept

The 1-NN algorithm partitions the input space $\mathcal{X}$ into N regions, based on the observation $\mathcal{D}$. For any input $x \in \mathcal{X}$, the predicted label is assigned according to the label of closest sample in $\mathcal{D}$, i.e. the region in which x belongs. This formally reads as the following set partition:

$$\mathcal{X} = \bigcup_{i=1}^{N} \left\{x \in \mathcal{X} | \forall x_l \in \mathcal{D},\ d\left(x, x_i\right) \leq d\left(x, x_l\right)\right\}. \tag{2.28}$$

14. Note that this imposes $\mathcal{X}$ to be a metric space, as the input space must possess a distance.



(a) (b)

Figure 2.8: Illustration of the Voronoï diagram concept ($\mathcal{X} = \mathbb{R}^2$) with the ℓ_2 distance: in (a) the regions are based on the inputs, i.e. Eq. (2.28), while (b) considers regions based on the labels (two possible labels, color-coded as red/blue), i.e. Eq. (2.29).

This partition is known as a *Voronoï diagram*, and depends on the distance design. This partition can also be seen from a label perspective: there will be as many regions as the number of different labels in $\mathcal{D}$. Letting $M \leq N$ be the number of different labels in $\mathcal{D}$, i.e. $M = \sum_{y \in \mathcal{Y}} \mathbb{1}_{\{y \in \mathcal{Y} | \exists (x_i, y_i) \in \mathcal{D}, y_i = y\}}(y)$, the partition rewrites as:

$$\mathcal{X} = \bigcup_{i=1}^{M} \left\{ x \in \mathcal{X} | y_{j^\star} = y_i, j^\star = \underset{j \in [\![1,N]\!]}{\arg\min}\, d(x, x_j) \right\}. \tag{2.29}$$

Both partitions are illustrated in Fig. 2.8.

Extension to k-NN

As opposed to 1-NN, the k-NN algorithm predicts a label based on the k closest neighbors' labels. As such, it usually provides a more reliable prediction. Indeed, if there exists labelization errors in $\mathcal{D}$, the prediction proposed by the 1-NN might be false: taking into account multiple neighbors allows to mitigate this issue. The decision process depends if the k-NN method is used for a regression or a classification task.

- **Regression.** In this setting, the predicted label for an input $x \in \mathcal{X}$ corresponds to a weighted linear combination of the labels of the k nearest neighbors in $\mathcal{D}$. A common approach is to use distance-based weights, i.e.:

$$\hat{y} = \sum_{i \in \mathcal{N}_k^{\mathcal{D}}(x)} \frac{1}{d(x, x_i)} y_i, \tag{2.30}$$

  where $\mathcal{N}_k^{\mathcal{D}}(x)$ denotes the set constituted by the indices of the k-nearest neighbors of x in $\mathcal{D}$.

- **Classification.** In this context, the predicted label for an input $x \in \mathcal{X}$ is given by

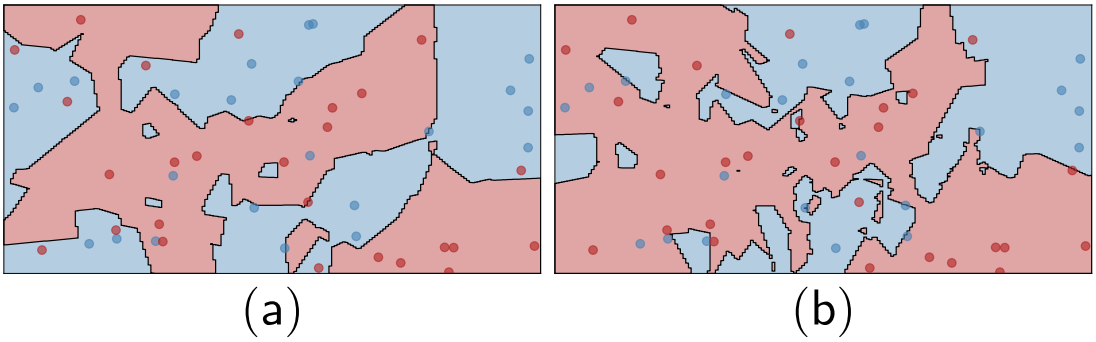


Figure 2.9: Illustration of the k-NN method in a binary classification context ($\mathcal{X} = \mathbb{R}^2$, $\mathcal{Y} = [0, 1]$) with the ℓ_2 distance: (a) $k = 3$, (b) $k = 8$. The background color at a given location designs the associated label based on the k-nearest neighbors in $\mathcal{D}$.

the majority label among the k-nearest neighbors of x in $\mathcal{D}$:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \sum_{y_i \in \mathcal{N}_k^{\mathcal{D}}(x)} \mathbb{1}_{\{y \in \mathcal{Y} | y_i = y\}}(y) \,. \tag{2.31}$$

It then becomes evident that the choice of the hyperparameter k is crucial: a small value of k may lead to a model that is sensitive to noise in the data, while a large k value tends to smooth out predictions but may overlook minority class labels due to averaging effects. An illustration of the effect of this hyperparameter in a binary classification context is proposed in Fig. 2.9, where the background color at given location represents the predicted class.

To summarize, k-NN is a non-parametric supervised learning method for both regression and classification tasks. Consequently, the output space depends on the considered task. Owing to its non-parametric nature, the k-NN method does not assume an explicit loss function but rather depends on a distance metric to identify neighbors in the input space, and does not admit an analytical expression. Common implementations of k-NN rely on computing a distance graph between each data point in the observation $\mathcal{D}$, which can be computationally intensive. Popular algorithms, such as K-D tree [Ben75] or ball tree approaches [Omo89], offer efficient search capabilities and perform well even in high-dimensional spaces.

2.3.3 Principal component analysis

PCA is an unsupervised learning method for dimensionality reduction tasks: a detailed presentation can be found in [GBC16, Chapter 2, pp. 48-52, Chapter 5, pp. 147-150], or [SSBD14, Chapter 23, pp. 324-326]. It is a very powerful method to perform data exploration or data pre-processing, with applications in many fields, e.g. genetics, computer vision, finance, or marketing. The objective of PCA is to identify a low-dimensional representation space in which the variance of the projected data is maximized along the

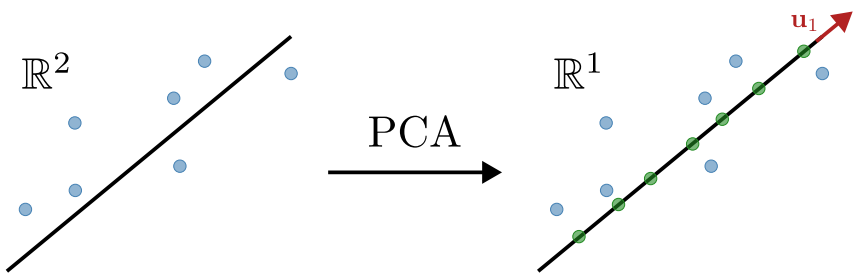


Figure 2.10: Illustration of PCA: the data, represented in the blue points in $\mathbb{R}^2$ are embedded on the $\mathbb{R}^1$ line as the green points, with $\mathbf{u}_1$ being the new space basis vector.

new basis vectors. More formally, given the observation $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^{D\times N}$, which can be rewritten as $\mathbf{X} \in \mathbb{R}^{D\times N}$, and assuming centered data [15], the PCA of $\mathbf{X}$ is a linear orthonormal transformation, mapping $\mathbf{X}$ to a d-dimensional subspace ($d < D$), and maximizing the variance of $\mathbf{X}$ along the subspace basis vectors. It can be expressed as the following optimization problem:

$$\begin{aligned} \underset{\mathbf{V}}{\text{maximize}} \quad & \frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \left\|\mathbf{V}^\mathsf{T}\mathbf{x}\right\|_2^2 . \\ \text{subject to} \quad & \mathbf{V} \in \mathbb{R}^{D\times d}, \\ & \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{Id}_d \end{aligned} \tag{2.32}$$

Here, the input space is $\mathcal{X} = \mathbb{R}^D$, the output space is $\mathcal{Y} = \mathbb{R}^d$, and the loss function is the squared ℓ_2 norm. It can be shown that this problem admits the following analytical solution:

$$\mathbf{V}^\star = \mathbf{U}_d = [\mathbf{u}_1, \cdots, \mathbf{u}_d] , \tag{2.33}$$

where $\mathbf{U}_d \in \mathbb{R}^{D\times d}$ are the eigenvectors associated to the d largest eigenvalues of the data covariance matrix $\mathbf{X}\mathbf{X}^\mathsf{T}$. A proof of this solution is proposed in Appendix B.2, and an illustration is provided in Fig. 2.10. The choice of the subspace dimension d is critical: a dimension that is too low may fail to capture the underlying complexity of the data, whereas a dimension that is too high may result in a representation that is difficult to interpret or exploit effectively.

2.4 Focus on deep learning

This section focuses on deep learning methods, which constitute the central ML methods used in current ML research.

15. Such that the observation is centered: $\sum_{i=1} \mathbf{x}_i = \mathbf{0}$.

2.4.1 Motivations

While the aforementioned ML methods present good performance across a broad range of problems, the recent surge of interest in ML is largely driven by the rise of *deep learning*, and, in particular, by the success of deep neural networks. Applications ranging from image recognition, protein structure prediction to speech recognition and conversational agents have showcased the versatility and effectiveness of modern neural networks. By definition, deep learning constitutes a subfield of ML that leverages deep neural architectures to address complex tasks. In contrast to traditional machine learning methods, which requires models to be tailored to the problem at hand, neural networks can adapt their representations directly from data. This capacity for data-driven adaptation constitutes the core advantage of neural networks and deep learning. A more formal definition of the deep learning paradigm, in a supervised learning setting, is presented in Highlight 2.3.

Highlight 2.3: Deep learning paradigm

Given an unknown mapping $\psi\colon \mathcal{X} \to \mathcal{Y}$, and a given error measure $\mu_{\mathcal{Y}}\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$, the objective of deep learning is to train a *neural network* $f_{\boldsymbol{\theta}}\colon \mathcal{X} \to \mathcal{Y}$ on data $\mathcal{D}$ such that, after training:

$$\mathbb{E}_{(x,\psi(x))\sim p_{x,y}} \left[\mu_{\mathcal{Y}} \left(f_{\boldsymbol{\theta}}\left(x\right), \psi\left(x\right)\right)\right] \leq \epsilon, \tag{2.34}$$

where the error level ϵ depends on the considered problem. The parameters $\boldsymbol{\theta}$ are called the *learnable parameters* of the network, and are updated during the training process. Note that the structure of $f_{\boldsymbol{\theta}}$ can be either very simple or very complex, depending on the complexity of the target mapping ψ.

Artificial neuron

The neural network field is not particularly recent, with work on a mathematical model of the neuron tracing back to 1943 [MP43]. In this work, McCulloch and Pitts drew inspiration from the biological description of a neuron to propose a mathematical model, thereby introducing one of the earliest forms of the artificial neuron. This model is formally introduced in Definition 2.9. While original research on neural networks aimed at mimicking the biological organization of mammals brains, current research only focus towards finding neural architectures that can learn complex functions, with no intent towards biological similarities.

Definition 2.9: Artificial neuron

Given an input $\mathbf{x} \in \mathbb{R}^n$, a weight vector $\mathbf{w} \in \mathbb{R}^n$, and an activation function $\sigma \colon \mathbb{R} \to \mathbb{R}$, such that $\sigma(\mathbf{x})_i = \sigma(x_i)$, the output of an artificial neuron is given by:

$$f_{\boldsymbol{\theta}} = \sigma\left(\mathbf{w}^\mathsf{T}\mathbf{x}\right), \tag{2.35}$$

where $\boldsymbol{\theta} = \mathbf{w}$. The weight vector and activation function allow the output to be a non-linear weighted combination of the inputs.

The first implementation of the neuron model was established by Rosenblatt in the late 1950s [Ros57]. The implemented model, known as the *perceptron*, consisted of a unique neuron equipped with a sign activation function, and was used to solve a binary classification task. However, while there exists some results regarding theoretical learning capabilities of the perceptron, e.g. Novikoff's convergence theorem [Nov62], its inherent simple structure limits its expressive power: a simple artificial neuron is only able to learn simple functions.

Multi-layer perceptron

The multi-layer perceptron (MLP) model can be viewed as an improved perceptron model. It consists of multiple stacked neurons, connected by non-linear functions. The inputs and outputs of the MLP are respectively called the *input* and *output layers*, while the successive stacked neurons layers are called the *hidden layers*. A formal definition of the MLP model is presented in Definition 2.10, and an illustration is provided in Fig. 2.11.

Definition 2.10: Multi-layer perceptron

For an input $\mathbf{x} \in \mathbb{R}^{n_0}$, the output of a MLP with p layers is given by:

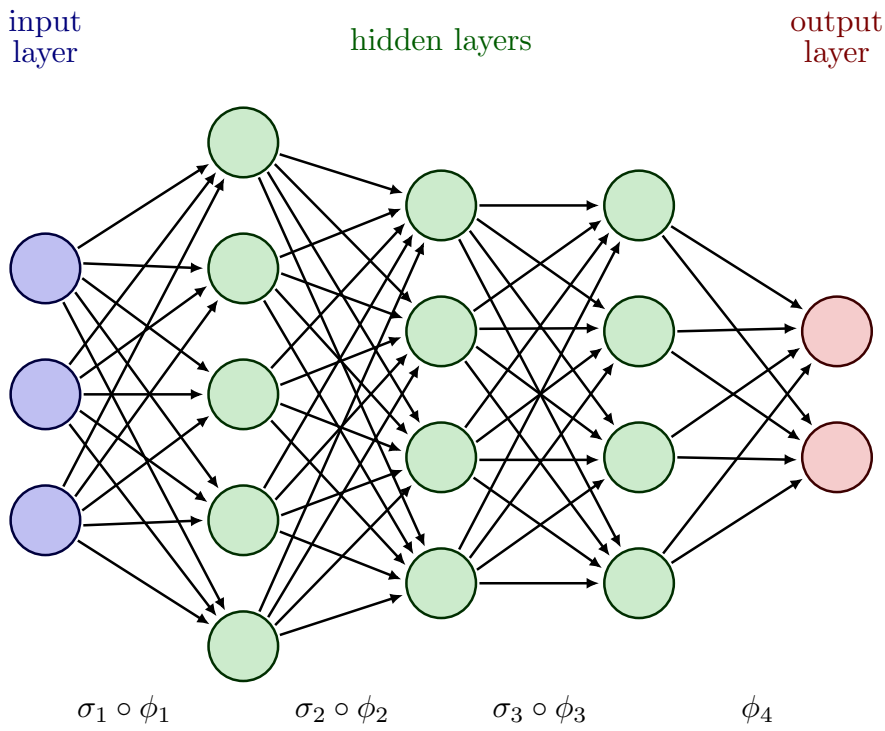
$$f_{\boldsymbol{\theta}}(\mathbf{x}) = (\phi_p \circ \sigma_{p-1} \circ \phi_{p-1} \cdots \circ \sigma_1 \circ \phi_1)(\mathbf{x}), \tag{2.36}$$

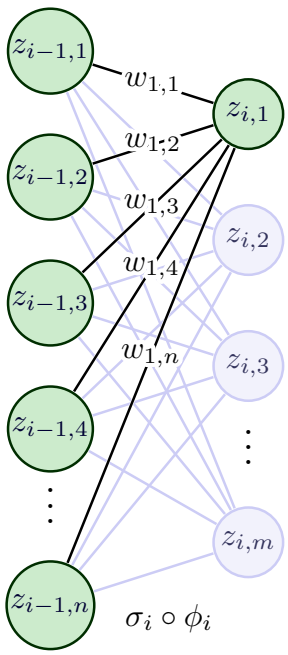
where $\sigma_i \colon \mathbb{R} \to \mathbb{R}$, such that $(\sigma_i \circ \mathbf{x})_j = \sigma_i \circ x_j$, and $\phi_i \colon \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$ respectively denote the activation function and affine mapping [a] of the ith layer. Specifically:

$$\phi_i(\mathbf{x}) = \mathbf{W}_i\mathbf{x} + \mathbf{b}_i, \tag{2.37}$$

with $\mathbf{W}_i \in \mathbb{R}^{n_i \times n_{i-1}}$, and $\mathbf{b}_i \in \mathbb{R}^{n_i}$.

a. Note that complex affine mappings can also be considered.





$$= \sigma_i \left(w_{1,1} z_{i-1,1} + w_{1,2} z_{i-1,2} + \cdots + w_{1,n} z_{i-1,n} + b_1 \right)$$

$$= \sigma_i \left(\sum_{j=1}^{n} w_{1,j} z_{i-1,j} + b_1 \right)$$

$$\begin{bmatrix} z_{i,1} \\ z_{i,2} \\ \vdots \\ z_{i,m} \end{bmatrix} = \sigma_i \left(\begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,1} & w_{2,2} & \dots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \dots & w_{m,n} \end{bmatrix} \begin{bmatrix} z_{i-1,1} \\ z_{i-1,2} \\ \vdots \\ z_{i-1,n} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \right)$$

$$\mathbf{z}_i = \sigma_i \left(\mathbf{W}_i \mathbf{z}_{i-1} + \mathbf{b}_i \right)$$

Figure 2.11: Illustration of a MLP neural network: the input lies in $\mathbb{R}^3$, there are 3 hidden layers, and the output lies in $\mathbb{R}^2$. The bottom schematic illustrates the behavior of an hidden layer: $\mathbf{z}_i = (\sigma_i \circ \phi_i)(\mathbf{z}_{i-1})$, ϕ_i being an affine mapping and σ_i a generic activation function (based on [Neu25b]).

This complex structure endows MLPs with high expressive power, i.e. MLPs are able to approximate a broad class of functions. By intuition, one can think that any given function can be approximated by a composition of non-linear functions: in the MLP structure, the affine mapping accounts for the representation of linear functions, while the activation functions introduce non-linearities. This intuition is formalized by a series of theorems, based on initial results of Cybenko, Hornik, Stinchcombe, and White [Cyb89; HSW89], showing that MLPs are universal function approximators. Given certain conditions on the activation functions, a MLP with at least one sufficiently wide hidden layer can approximate any continuous function [GBC16, Chapter 6, pp. 198-201]. A more formal definition is presented in Theorem 2.1.

Theorem 2.1: Universal approximation

Let $C(\mathcal{X}, \mathcal{Y})$ be the set of continuous functions from $\mathcal{X}$ to $\mathcal{Y}$. Let $n \in \mathbb{N}^*$, $m \in \mathbb{N}^*$, $\sigma \in C(\mathbb{R}, \mathbb{R})$, and $\mathcal{K}$ be a compact subset of $\mathbb{R}^n$, $\mathcal{K} \subseteq \mathbb{R}^n$. Given $\psi \in C(\mathcal{K}, \mathbb{R}^m)$, $\forall \epsilon > 0$, $\exists k \in \mathbb{N}^*$, $\mathbf{W}_1 \in \mathbb{R}^{k \times n}$, $\mathbf{b}_1 \in \mathbb{R}^k$, $\mathbf{W}_2 \in \mathbb{R}^{m \times k}$, $\mathbf{b}_2 \in \mathbb{R}^m$ such that:

$$\sup_{\mathbf{x} \in \mathcal{K}} \|\psi(\mathbf{x}) - \mathbf{W}_2(\sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2)\|_2 \leq \epsilon. \tag{2.38}$$

In other words, the set of functions representable by MLPs is dense in $C(\mathcal{K}, \mathbb{R}^m)$: any continuous function of a compact subspace of $\mathbb{R}^n$ can be approximated by a two-layer MLP with one arbitrarily wide hidden layer.

An alternative version of this theorem exists for a fixed layer width, i.e. number of neurons per layer, but arbitrarily deep network [16], i.e. number of layers [Lu+17]. Recent research focuses on finding bounds for the expressive power of MLPs with fixed layer width and depth: an interested reader is referred to [Par+21].

Neural network training

Although the universal approximation theorem guarantees the existence of a MLP capable of approximating virtually any function, it does not provide information regarding the needed structure, i.e. which depth and width to choose, nor on the training procedure necessary to determine the parameters of each layer for a given task. However, owing to their compositional structure, MLPs can be efficiently trained through gradient descent, using efficient differentiation techniques such as backpropagation [RHW86] [17].

It is proposed to illustrate this concept through a supervised learning problem involving the generic neural network $f_{\boldsymbol{\theta}}(\mathbf{x}) = \left(f_{\boldsymbol{\theta}_p} \circ f_{\boldsymbol{\theta}_{p-1}} \circ \cdots \circ f_{\boldsymbol{\theta}_1}\right)(\mathbf{x})$, where $f_{\boldsymbol{\theta}_i}$ denotes the

16. The depth of a neural network is sometimes denoted as its width.
17. An interested reader is referred to [GBC16, Chapter 6, pp. 204-227] for additional information.

ith layer with learnable parameters $\boldsymbol{\theta}_i$. In this setting, the empirical risk minimization problem reads as:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{minimize}} \quad & \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \mu\left(\mathbf{y}, f_{\boldsymbol{\theta}}\left(\mathbf{x}\right)\right), \\ \text{subject to} \quad & \boldsymbol{\theta} \in \mathcal{H}_\theta \end{aligned} \tag{2.39}$$

where $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ is the observation, and $\mathcal{H}_\theta$ is the learnable parameters hypothesis class. An approximate solution can be found using gradient descent [18]. The learnable parameters of each layers are updated as:

$$\forall i \in [\![1, p]\!],\ \boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \lambda \nabla_{\boldsymbol{\theta}_i} \mu\left(\mathbf{y}, f_{\boldsymbol{\theta}}\left(\mathbf{x}\right)\right). \tag{2.40}$$

The complexity resides in the computation of $\nabla_{\boldsymbol{\theta}_i} \mu\left(\mathbf{y}, f_{\boldsymbol{\theta}}\left(\mathbf{x}\right)\right)$. However, owing to the compositional structure of neural networks, it is possible to use the chain's rule to easily compute each gradient, as presented in Definition 2.11.

Definition 2.11: Chain's rule for backpropagation

Let $\mathbf{x} \in \mathbb{R}^{n_0}$, and $\mu \colon \mathbb{R}^{n_p} \times \mathbb{R}^{n_p} \to \mathbb{R}^+$. Let $f_{\boldsymbol{\theta}}\left(\mathbf{x}\right) = \left(f_{\boldsymbol{\theta}_p} \circ f_{\boldsymbol{\theta}_{p-1}} \circ \cdots \circ f_{\boldsymbol{\theta}_1}\right)\left(\mathbf{x}\right)$, such that:

$$f_{\boldsymbol{\theta}_i} \colon \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}, \forall i \in [\![1, p]\!] \tag{2.41}$$

$$\mathbf{z}_i = f_{\boldsymbol{\theta}_i}\left(\mathbf{z}_{i-1}\right), \forall i \in [\![1, p]\!] \tag{2.42}$$

$$\mathbf{z}_0 = \mathbf{x}. \tag{2.43}$$

Let $\gamma = \mu\left(\mathbf{y}, f_{\boldsymbol{\theta}}\left(\mathbf{x}\right)\right)$. The gradient of the jth parameter of the ith layer reads as:

$$\frac{\partial}{\partial \theta_{i,j}} \gamma = \sum_{k_p=1}^{n_p} \frac{\partial \mu\left(\mathbf{y}, \mathbf{z}_p\right)}{\partial z_{p,k_p}} \frac{\partial z_{p,k_p}}{\partial \theta_{i,j}} \tag{2.44}$$

$$= \sum_{k_p=1}^{n_p} \sum_{k_{p-1}=1}^{n_{p-1}} \cdots \sum_{k_i=1}^{n_i} \frac{\partial \mu\left(\mathbf{y}, \mathbf{z}_p\right)}{\partial z_{p,k_p}} \frac{\partial z_{p,k_p}}{\partial z_{p-1,k_{p-1}}} \cdots \frac{\partial z_{i+1,k_{i+1}}}{\partial z_{i,k_i}} \frac{\partial z_{i,k_i}}{\partial \theta_{i,j}}. \tag{2.45}$$

Introducing the Jacobian matrix of $f \colon \mathbb{R}^n \to \mathbb{R}^m$, s.t. $\mathbf{y} = f\left(\mathbf{x}\right)$, as $\Delta_{\mathbf{y},\mathbf{x}} = [\nabla_{x_1}\mathbf{y}, \cdots, \nabla_{x_n}\mathbf{y}] \in \mathbb{R}^{m\times n}$, the total gradient of the ith layer parameters can be expressed as:

$$\nabla_{\boldsymbol{\theta}_i} \mu\left(\mathbf{y}, f_{\boldsymbol{\theta}}\left(\mathbf{x}\right)\right) = \Delta^{\mathsf{T}}_{\mathbf{z}_i,\boldsymbol{\theta}_i} \left(\prod_{k=i+1}^{p} \Delta^{\mathsf{T}}_{\mathbf{z}_k,\mathbf{z}_{k-1}}\right) \nabla_{\mathbf{z}_p} \mu\left(\mathbf{y}, \mathbf{z}_p\right). \tag{2.46}$$

18. See Section 2.2.2.

$$\mathbf{x} \xrightarrow{f_{\boldsymbol{\theta}_1}} \mathbf{z}_1 \xrightarrow{f_{\boldsymbol{\theta}_2}} \cdots \xrightarrow{f_{\boldsymbol{\theta}_i}} \mathbf{z}_i \xrightarrow{f_{\boldsymbol{\theta}_{i+1}}} \cdots \xrightarrow{f_{\boldsymbol{\theta}_{p-1}}} \mathbf{z}_{p-1} \xrightarrow{f_{\boldsymbol{\theta}_p}} \mathbf{z}_p \xrightarrow{\mu} \gamma = \mu\left(\mathbf{y}, f_{\boldsymbol{\theta}}\left(\mathbf{x}\right)\right)$$

$$\nabla_{\boldsymbol{\theta}_p}\gamma = \Delta^{\mathsf{T}}_{\mathbf{z}_p,\boldsymbol{\theta}_p}\nabla_{\mathbf{z}_p}\gamma$$

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_{p-1}}\gamma &= \Delta^{\mathsf{T}}_{\mathbf{z}_{p-1},\boldsymbol{\theta}_{p-1}}\underbrace{\Delta^{\mathsf{T}}_{\mathbf{z}_p,\mathbf{z}_{p-1}}}_{\boldsymbol{\Phi}_{p-1}}\nabla_{\mathbf{z}_p}\gamma \\ &= \Delta^{\mathsf{T}}_{\mathbf{z}_{p-1},\boldsymbol{\theta}_{p-1}}\boldsymbol{\Phi}_{p-1}\nabla_{\mathbf{z}_p}\gamma\end{aligned}$$

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_i}\gamma &= \Delta^{\mathsf{T}}_{\mathbf{z}_i,\boldsymbol{\theta}_i}\underbrace{\Delta^{\mathsf{T}}_{\mathbf{z}_{i+1},\mathbf{z}_i}\cdots\Delta^{\mathsf{T}}_{\mathbf{z}_{p-1},\mathbf{z}_{p-2}}\Delta^{\mathsf{T}}_{\mathbf{z}_p,\mathbf{z}_{p-1}}}_{\boldsymbol{\Phi}_i}\nabla_{\mathbf{z}_p}\gamma \\ &= \Delta^{\mathsf{T}}_{\mathbf{z}_i,\boldsymbol{\theta}_i}\boldsymbol{\Phi}_i\nabla_{\mathbf{z}_p}\gamma\end{aligned}$$

Figure 2.12: Illustration of chain's rule for $f_{\boldsymbol{\theta}}\left(\mathbf{x}\right) = \left(f_{\boldsymbol{\theta}_p} \circ f_{\boldsymbol{\theta}_{p-1}} \circ \cdots \circ f_{\boldsymbol{\theta}_1}\right)\left(\mathbf{x}\right)$: the gradient update for the parameters of the ith layer requires computing the updates for layers $i+1, \cdots, p$.

The gradient of the parameters at an arbitrary layer i, presented in Eq. (2.46), can be decomposed into three parts:

- $\Delta^{\mathsf{T}}_{\mathbf{z}_i,\boldsymbol{\theta}_i}$ describes the gradients at the ith layer, i.e. how a small change in parameters $\boldsymbol{\theta}_i$ affects the layer's output $\mathbf{z}_i$.
- $\prod_{k=i+1}^{p}\Delta^{\mathsf{T}}_{\mathbf{z}_k,\mathbf{z}_{k-1}}$ represents how the gradient of the loss propagates backward from the output layer to the ith layer, capturing the contribution of subsequent layers to the update of the parameters at layer i.
- $\nabla_{\mathbf{z}_p}\mu\left(\mathbf{y}, \mathbf{z}_p\right)$ represents the loss function contribution to the parameters update.

Additionally, one can remark a recursive structure in the Jacobian products. Introducing $\boldsymbol{\Phi}_i = \prod_{k=i+1}^{p}\Delta^{\mathsf{T}}_{\mathbf{z}_k,\mathbf{z}_{k-1}}$, with $\boldsymbol{\Phi}_p = \mathbf{Id}$, one has:

$$\boldsymbol{\Phi}_i = \Delta^{\mathsf{T}}_{\mathbf{z}_{i+1},\mathbf{z}_i}\boldsymbol{\Phi}_{i+1}. \tag{2.47}$$

Computing $\nabla_{\boldsymbol{\theta}_i}\mu\left(\mathbf{y}, f_{\boldsymbol{\theta}}\left(\mathbf{x}\right)\right)$ is thus relatively straightforward in principle, as the parameter update at layer i relies on gradients computed for layer $i+1$. However, it can entail significant computational complexity in practice. First, depending on the depth of the neural network, updating the parameters of the first layer requires propagating gradients through all subsequent layers. This process can be computationally demanding and may lead to vanishing gradients in networks with a large number of layers. Second, this process relies on the ability to compute the gradient of each layer's output with respect to its input,

meaning that all operations in the network must be differentiable. An illustration of the backpropagation concept is proposed in Fig. 2.12.

Remark 2.4. *Gradient descent is leveraged to solve the optimization problem described Eq.* (2.39)*. In order to obtain an approximate solution with a reasonably low approximation error, the loss function should be convex with respect to the learnable parameters. However, due to the complex structure of neural networks, this convexity condition is rarely satisfied. Despite this, gradient descent and its variants form the core of modern learning methods, with the hope of converging towards a sufficiently good local minimum.*

2.4.2 Popular architectures

While MLPs were among the first neural architectures, modern neural networks exhibit a higher degree of sophistication while still following the MLP principles: they usually consist of input and output layers, with one or more intermediate hidden layers. It is important to note that the no free lunch theorem still applies to neural networks: no single architecture can achieve high performance across all learning tasks or data modalities. Consequently, the objective is to select an architecture that is appropriate for the specific task at hand, and to train it using the available data. Some popular architectures are presented in the following paragraphs.

Convolutional networks

Convolutional neural networks (CNNs) [LeC+89] were originally introduced for image processing tasks but are applicable to a wide range of pattern recognition problems. Their design is motivated by the observation that semantic content of an image is invariant to translations. This prior knowledge naturally motivates the use of convolution operators, which enforce translation-equivariance in the learned representations, i.e. the property that a translation of the input results in an equivalent translation of the output. A basic CNN architecture thus consists of several convolutional layers that extract local structures from the input data, followed by a MLP that combines these features to produce the final output. CNNs can either solve classification problems, e.g. determining whether an image depicts a cat or a dog, or regression problems, e.g. predicting a car's retail price from its image. It is worth noting that *CNNs solved the image recognition task*: modern CNN architectures outperform humans in image recognition problems. An interested reader is referred to [GBC16, Chapter 9, pp. 330-372] for more information.

Autoencoders

Autoencoders (AEs) are a class of neural networks designed to address representation learning tasks. Specifically, they consist of two main components: an *encoder*, which maps the input data to a lower-dimensional space [19], and a *decoder*, which maps the encoded

19. Note that, in the context of AEs, the low-dimensional space is also denoted as the *latent space*.

data back to the input space. The network is trained in a end-to-end manner with the goal of minimizing the difference between the input and the output. As such, AEs are trained in an unsupervised manner: by definition, the training does not rely on labeled data. Once trained, the learned encoded representation can be used for several tasks, e.g. data compression, structure discovery, or denoising. Note that the contribution presented in Chapter 6 is based on an AE structure. An interested reader is referred to [GBC16, Chapter 14, pp. 502-525] for more information.

Transformers

Transformer networks [Vas+17] constitute the core architecture of large language models that are used in conversational agents such as ChatGPT. This neural architecture relies heavily on the *attention mechanism*, which, given an input sequence, computes a score for each element, reflecting its relative importance with respect to the others. This enables the neural network to capture the contextual relevance of each input element and leverage this information to perform the required task. Transformer networks have been successfully used in a broad class of tasks such as natural language generation [Bro+20], speech recognition [Gul+20] or image generation [PX23].

2.4.3 Known limitations

Over the past decade, neural networks have achieved impressive results across a wide range of tasks. However, several limitations have emerged, some of which are outlined below.

Training difficulties

As for every ML method, and especially given the complexity of the problems they address, neural networks require a training phase that can be extensive. This is especially evident in modern neural architectures, such as those powering conversational agents, which often contain a very large number of learnable parameters, that must all be updated during training. For instance, GPT-3 posses 175 billion learnable parameters [Bro+20], and it is estimated that GPT-4 posses around 1.7 trillion learnable parameters. Such complex architectures raise several issues:

- **Data requirements.** By definition, neural networks, and more generally machine learning methods, are data-driven, learning patterns directly from observed data. Consequently, depending on the complexity of the task at hand, a substantial amount of data is often required to achieve satisfactory performance. As an example, the dataset used to train GPT-3 comprises nearly 500 billion tokens [Bro+20], corresponding to approximately 375 billion worlds. If this content were printed on standard A4 paper in 12-point Times New Roman font, it would occupy roughly 937 million pages. When stacked, these pages would reach an approximated height

of 94 kilometers, ten times the height of Mount Everest, and nearly the edge of space (Kármán line). Recent research estimates that if large language models continue their developments trend, they will be trained on the totality of public human generated text data between years 2026 and 2032 [Vil+24], as illustrated in Fig. 3.

- **Memory requirements.** Modern neural networks are typically trained on GPUs, which allow for significantly faster training times compared to conventional central processing units (CPUs). However, each GPU has physical memory constraints, imposing the use of complex distributed architectures to train large networks. For instance, it is estimated that GPT-4 has been trained on 20000 NVIDIA GPUs.

- **Training time.** Since every learnable parameter must be updated during training, and because the loss function with respect to the learnable parameters generally becomes more chaotic as their number increases, the training time tends to grow with the number of learnable parameters. For instance, it is estimated that GPT-4 required around three months of continuous training, with a revealed cost reaching around 100 million USD.

Note that examples with the GPT-3/4 models represent extreme cases, as these models embody the highest levels of complexity in current neural architectures. Nonetheless, they are indicative of the challenges in training neural networks.

Interpretability

One of the main criticism of deep learning lies in its limited interpretability. Indeed, while a trained neural network can be viewed as a function mapping an input to an output, its analytical form is typically a composition of many layers, offering little to no insight into the nature of the learned mapping. As a result, some complex neural architectures are often referred to as *black boxes*, due to the lack of transparency in their decision-making process. Current research in the deep learning field tries to propose neural architectures with increased interpretability. Examples illustrating this research effort are Kolmogorov-Arnold networks (KANs) [Liu+25], and the fields of physics-informed neural networks (PINNs), and model-based machine learning (MB-ML). These research directions are rapidly presented below:

- **KANs.** The core idea behind KANs is the use of the Kolmogorov-Arnold representation theorem [SchHie21] to structure a neural network. The associated structure is still organized within layers, but only learns non-linear activation functions. It was shown in [Liu+25] that, although KANs may require longer training times compared to MLPs, their resulting structure offers significantly improved interpretability.

- **PINNs and MB-ML.** These approaches aim to incorporate physical laws or a priori problem knowledge directly into the learning process. MB-ML forms the core of Chapter 3, where a detailed comparison between MB-ML and PINNs will be provided.

2.5 Conclusion

This chapter aimed at providing an introduction to the machine learning field. The main takeaways are summarized below.

- Machine learning is a general field focused on developing methods that learn from data. It encompasses several paradigms, including supervised learning, unsupervised learning, or reinforcement learning. Within these paradigms, various methods exist to solve tasks such as regression, classification, or dimensionality reduction.
- The core idea of machine learning is to solve an empirical risk minimization problem, where a learning algorithm optimize parameters of a model, over training data, to solve a given task. Additionally, different model selection criteria guard against overfitting.
- Deep learning constitutes a subfield of machine learning that uses deep neural networks in learning tasks. Neural networks consist of composed parameterized functions whose parameters are optimized to minimize task-specific loss functions.
- There exists many different neural network architectures, each designed for particular learning tasks.
- While neural networks present impressive performance in many tasks, they present limitations such as complex training procedure, long training time or lack of interpretability. These limitations can be mitigated, for instance through the use of the model-based machine learning paradigm, which is introduced in the following chapter.

Chapter 3

Model-based machine learning

This chapter aims at providing an overview of the model-based machine learning paradigm, and see how it fundamentally differs from classical machine learning methods. First, the general mathematical framework of inference rule design is presented in Section 3.1, as well as how model-based and machine learning methods can be incorporated within this framework. Then, Section 3.2 presents how it is possible to combine strengths of both the model-based and machine learning paradigms to increase performance, interpretability and reduce training complexities. Subsequently, Section 3.3 presents how the model-based machine learning paradigm can be applied to applications related to the wireless propagation channel. Finally, Section 3.4 provides some concluding remarks.

Contents

3.1 General context **80**
- 3.1.1 Inference rule design 80
- 3.1.2 Model-based methods 83
- 3.1.3 Machine learning methods 87

3.2 Towards model-based machine learning **90**
- 3.2.1 Structure 90
- 3.2.2 Optimize 94
- 3.2.3 Initialize 98

3.3 Application to the wireless channel **99**
- 3.3.1 Radio environment compression and localization 100
- 3.3.2 Joint parametric and hardware impairments estimation 102
- 3.3.3 Channel compression through channel charting 104
- 3.3.4 Use-case summary 107

3.4 Conclusion **108**

This chapter draws significant inspiration from the excellent book of Shlezinger and Eldar [SE23], which serves as a longer version of several foundational papers: [Shl+21; SEB22; Shl+23]. The book provides a rigorous and structured presentation of the model-based machine learning paradigm, supported by theoretical developments and valuable illustrative examples.

3.1 General context

Every data-processing problem can generally be approached through two main paradigms: *data-driven* and *model-based*. In the data-driven approach, solutions of the given problem are obtained by optimizing learnable parameters of a model-agnostic method, relying solely on the information provided by the available data. This constitutes the core of the machine learning framework, presented in details in Chapter 2. The main advantage of this approach is the absence of domain-specific knowledge requirements: a ML model can be trained without any a priori information about the considered data-processing problem. In particular, the adaptability of modern neural networks allows a single architecture to achieve satisfactory performance across a wide range of problems. However, this expressive power comes at the expense of high training complexity and limited interpretability [MLE21]. These architectures often demand large amounts of training data, long training times, and typically act as *black boxes*, since their outputs cannot be easily related to the provided inputs. On the other hand, the model-based approach relies on domain knowledge: using a priori information about the problem, and suitable simplifying assumptions allows the derivation of a mathematical model describing the problem. This often enables the formulation of a closed-form solution, or motivates the use of iterative algorithms to solve the problem, both of which are typically of relatively low computational complexity. However, owing to their reliance on simplifying assumptions, model-based methods are generally prone to high biases: when these assumptions hold, satisfactory performance can be achieved, but when they do not, a significant performance degradation is to be expected.

> **Highlight 3.1: Model-based machine learning paradigm**
>
> The philosophy of the model-based machine learning paradigm lies in combining elements of model-based approaches with machine learning techniques, thereby leveraging the advantages of both.

3.1.1 Inference rule design

From a theoretical perspective, model-based and machine learning methods are specific instances of the inference rule design framework. In this setting, the problem consists in finding an *inference rule* [1] f, mapping a *context* $x \in \mathcal{X}$ to a *decision* $y \in \mathcal{Y}$ [2]:

$$f : \mathcal{X} \to \mathcal{Y}, \tag{3.1}$$

1. Note that it is sometimes also called inference mapping, decision rule or decision mapping.
2. Note that, in [SE23], the decision is denoted by s, and the decision space by $\mathcal{S}$. It is proposed to adopt the y and $\mathcal{Y}$ notations to be consistent with the usual machine learning notations presented in Chapter 2.

where $\mathcal{X}$ denotes the context space and $\mathcal{Y}$ represents the decision space.

The inference rule design framework takes its name from its underlying philosophy: given a specific context x, the selected decision rule f allows to infer a decision $f(x)$. One can directly remark that the challenge lies is efficiently selecting the inference rule, as different tasks usually require different mappings. Subsequent paragraphs present each step of this paradigm in detail.

Highlight 3.2: Inference rule design philosophy

For a given problem, designing an inference rule can be decomposed into three distinct steps [SE23, Chapter 2, pp. 301 - 305]:

1. Select the inference rule type $\mathcal{H}$ adapted to the specific problem.
2. Tune the selected rule f with respect to a given error measure.
3. Evaluate the tuned inference rule.

Remark 3.1. *An experienced reader might notice that the inference rule design framework appears very similar to the machine learning paradigm introduced in Chapter 2. However, it constitutes an unifying framework that bridges the model-based and data-driven approaches. Indeed, this framework is sufficiently general to encompass a wide range of problems, including classical signal processing tasks such as estimation, denoising, or super-resolution, as well as control problems involving state-space evolution, but also machine learning problems such as regression or classification.*

Inference rule type selection

From a general perspective, any map belonging to the set of mapping from $\mathcal{X}$ to $\mathcal{Y}$, i.e. $\mathcal{Y}^{\mathcal{X}}$, constitutes a candidate mapping. In practice, testing every map from this set entails an intractable complexity. Therefore, the candidate inference rules belong to a restriction of this set, defined as:

$$\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}. \tag{3.2}$$

$\mathcal{H}$ is known as the inference rule type but is more usually defined as the hypothesis class. Usual hypothesis classes include linear models, neural networks, or iterative algorithms. These hypothesis classes can usually consists of parameterized functions, where each of these functions' parameters belong to a parameter space $\mathcal{H}_{\boldsymbol{\theta}}$, defined such that:

$$\mathcal{H} = \{f_{\boldsymbol{\theta}} | \boldsymbol{\theta} \in \mathcal{H}_{\boldsymbol{\theta}}\}. \tag{3.3}$$

In this context, $f_{\boldsymbol{\theta}}$ represents an inference rule parameterized by the parameters $\boldsymbol{\theta}$. As an example, for linear models, the hypothesis and parameter space classes are defined as:

$$\mathcal{H} = \left\{ f_{\boldsymbol{\theta}} \in \mathcal{Y}^{\mathcal{X}} | f_{\boldsymbol{\theta}} \colon \mathbf{x} \to \mathbf{A}\mathbf{x} + \mathbf{b} \right\} \tag{3.4}$$

$$\mathcal{H}_{\boldsymbol{\theta}} = \left\{ \mathbf{A}, \mathbf{b} | \mathbf{A} \in \mathbb{C}^{\dim_{\mathbb{R}}(\mathcal{Y}) \times \dim_{\mathbb{R}}(\mathcal{X})}, \mathbf{b} \in \mathbb{C}^{\dim_{\mathbb{R}}(\mathcal{Y})} \right\}. \tag{3.5}$$

Parameter tuning

The relevance of a given inference rule is evaluated using an error measure μ, defined as the following map [SE23, Section 2.3, pp. 303 - 304]:

$$\mu \colon \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{+}. \tag{3.6}$$

This measure characterizes the relevance of a rule f for a given context x and associated decision y. From this error measure, it is then possible to find the optimal inference rule within the hypothesis class, a task that can be formalized as the following optimization problem:

$$\begin{aligned} \underset{f_{\boldsymbol{\theta}}}{\text{minimize}} \quad & \mu\left(f_{\boldsymbol{\theta}}, x, y\right). \\ \text{subject to} \quad & f_{\boldsymbol{\theta}} \in \mathcal{H}, \\ & (x, y) \in \mathcal{X} \times \mathcal{Y} \end{aligned} \tag{3.7}$$

This optimization problem can also be rewritten from a parameter space perspective:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{minimize}} \quad & \mu\left(f_{\boldsymbol{\theta}}, x, y\right). \\ \text{subject to} \quad & \boldsymbol{\theta} \in \mathcal{H}_{\boldsymbol{\theta}}, \\ & (x, y) \in \mathcal{X} \times \mathcal{Y} \end{aligned} \tag{3.8}$$

Tuning an inference rule therefore consists in solving the optimization problem defined in Eq. (3.7), or equivalently in Eq. (3.8). The selection of the error measure is of paramount importance as it influences the tuned inference rule: if the error measure is not adapted to the given problem, the tuned rule will perform poorly. Usual error measures include the ℓ_1 and ℓ_2 norms, the conditional cross entropy, or the Kullback-Leibler divergence.

Inference rule evaluation

The final step of the presented framework consists in selecting an evaluation metric, and assessing the performance of the tuned inference rule. Indeed, as explained in [SE23, Section 2.3, pp. 304], the error measure used during the tuning step can be a surrogate of the actual problem objective. This originates from multiple reasons, some of which are listed below:

- **Complexity limitations.** The problem at hand may be too complex to define a clear objective that fully captures the desired outcome. This is particularly the case

in tasks addressed by modern deep learning methods, which are often so intricate that the error measure itself is defined as a combination of multiple criteria, each targeting a specific subproblem.

- **Regularizers.** The chosen error measure might include regularizers that do not directly constitute the core problem objective, but serve to obtain a tuned rule with certain properties. For instance, in sparse recovery problems [DE03; Fuc04; Don06; CRT06; TW10], sparsity-promoting regularizers, e.g. the ℓ_1 norm, are commonly used to enforce sparsity in the resulting solutions.

As a result, the error measure used during the tuning process may not fully capture the ultimate objective of the considered task. Therefore, it is often necessary to rely on a different evaluation metric to properly assess the actual performance of the tuned rule.

Having introduced the inference rule design framework, it is now proposed to present the relationship between this framework and the model-based and machine learning paradigms.

3.1.2 Model-based methods

The core idea behind model-based methods lies in the use of *domain knowledge* to define an analytical model describing the problem, which allows to ease the inference rule type selection and tuning steps. A definition of the domain knowledge concept is proposed in Definition 3.1.

Definition 3.1: Domain knowledge

For a given data-processing problem, domain knowledge denotes any a priori information pertaining to the problem. Such knowledge encompasses structural or statistical properties of the underlying model, as well as general physical constraints inherent to the problem setting.

As an illustrative example, it is proposed to consider the channel estimation problem, in which the objective is to infer the channel matrix from received measurements. In this context, domain knowledge encompasses information about the antenna system parameters, such as the antenna type, carrier frequency, or bandwidth, as well as characteristics of the propagation environment, including mobility conditions and whether the setting is urban or rural. It may also include knowledge of the system design, such as the type of emitted signals, and, if random, their underlying distributions.

Domain knowledge can be incorporated to facilitate the inference design steps in different non-exclusive ways, as detailed below:

- **Hypothesis class specification.** Generally, domain knowledge enables the derivation of an analytical model that can be used to obtain a solution of the problem at hand,

and that is valid under certain specific assumptions. This allows to restrict the hypothesis class to a subset of $\mathcal{Y}^{\mathcal{X}}$, thereby greatly reducing the complexity of the tuning step in the inference rule design framework. Note that if domain knowledge is sufficiently precise, the hypothesis class can be restricted to a single fixed mapping, bypassing the tuning step. Additionally, if the formulated model is random, domain knowledge generally allows to explicit the underlying distributions.

- **Error measure design.** As explicited in Eq. (3.7) and Eq. (3.8), the tuning step of the inference rule design relies on an error measure: domain knowledge often allows to select a task-adapted measure, such that the tuned inference rule present satisfactory performance.

Main model-based methods types

From a general perspective, model-based methods can be categorized within the *random* and *deterministic* settings [3]. The random setting is characterized by the presence of at least one random parameter within the analytical model used to solve the problem. Formally, the context x and the decision y are related through a joint distribution $p_{x,y}$. Consequently, the decision is not fully characterized by the context, as it may also be influenced by unknown random parameters. This setting is common in signal processing, where complex problems are often addressed using simplified deterministic analytical models augmented with random variables to capture phenomena that cannot be easily modeled otherwise. On the other hand, in the deterministic setting, the underlying model fully characterizes the decision based on the context, without involving any random variables.

Random setting. In this setting, domain knowledge of the underlying probability distributions can be used to tune the parameters of a parameterized function $f_{\boldsymbol{\theta}}$ that accounts for random elements. Specifically, given an hypothesis class $\mathcal{H}$, error measure μ, the selected rule is the solution of the following optimization problem:

$$\begin{aligned} \underset{f_{\boldsymbol{\theta}}}{\text{minimize}} \quad & \mathbb{E}_{(x,y)\sim p_{x,y}} \left[\mu\left(f_{\boldsymbol{\theta}}, x, y\right)\right], \\ \text{subject to} \quad & f_{\boldsymbol{\theta}} \in \mathcal{H} \end{aligned} \tag{3.9}$$

where $p_{x,y}$ denotes the underlying problem distribution. One can remark that this constitutes a risk minimization problem, as defined in Definition 2.5. Additionally, the chosen rule is the one that minimizes the mean error measure over the weighted context-decision pairs.

It is worth noting that the obtained rule still depends on the selected error measure. As an example it can be shown that, when selecting the squared ℓ_2 distance as the error measure, the obtained rule when solving the problem described in Eq. (3.9) is called the

3. Interested readers are referred to [SE23, Section 3.1, pp. 307-313], which further distinguishes between Bayesian and non-Bayesian methods within the random setting.

minimum mean squared error, whose expression is given by:

$$f_{\mathsf{MMSE}}(x) = \arg\min_{f \in \mathcal{H}} \mathbb{E}_{(x,y)\sim p_{x,y}} \left[\|y - f(x)\|_2^2 \right] \tag{3.10}$$

$$= \mathbb{E}_{y \sim p_{y|x}} [y|x] . \tag{3.11}$$

The selected rule is then defined as the conditional expectation of the decision y knowing the context x [4]. Other random methods also include the well-known maximum a priori or maximum likelihood methods. For a detailed discussion on the impact of the choice of the error measure in the risk minimization problem, an interested reader is referred to [SE23, Section 3.1, pp. 307 - 308].

Deterministic setting. The deterministic case can be viewed as a special case of the random setting, where the decision y is fully determined by the context x via an unknown function ψ, i.e.:

$$y = \psi(x) . \tag{3.12}$$

This implies that the joint distribution $p_{x,y}$ degenerates to:

$$p_{x,y}(x, y) = p_x(x)\, \delta(y - \psi(x)) , \tag{3.13}$$

where $p_x(x)$ denotes the context probability density function, and $\delta(y - \psi(x))$ represents the deterministic conditional distribution $p_{y|x}$, reflecting the absence of randomness in the decision y given the context x. As such, both settings can be framed within the same risk minimization paradigm, albeit with distinct objective functions. Particularly, in the deterministic setting, domain knowledge can be leveraged to define a task-specific error measure, and to derive an analytical model that can be used to obtain a solution to the problem at hand, thereby restricting the hypothesis class.

The regularized least-squares method constitutes a perfect example of this approach. Domain knowledge of the problem is leveraged to determine an analytical model $g\colon \mathcal{Y} \to \mathcal{X}$, which maps a decision to a context. This model is then used to select the decision corresponding to a given context:

$$f_{\mathsf{RLS}}(x) = \arg\min_{y \in \mathcal{Y}} \|x - g(y)\|_2^2 + \phi(y) . \tag{3.14}$$

Furthermore, domain knowledge is also incorporated into the design of a regularizer ϕ, that can enable the promotion of sparsity, the enforcement of physical constraints, or the incorporation of other problem-specific properties.

Deriving a solution

In both random and deterministic settings, the obtained inference rule often takes the form of an optimization problem that must be solved. This problem can either admit an

4. A demonstration of this expression is proposed in Appendix C.1.

analytical solution, or be solved using iterative solvers such as gradient descent, proximal gradient descent, or alternating direction of multipliers [5]. It should be noted that certain tasks may yield inference rules that can be expressed as optimization problems for which no analytical solution exists, and that cannot be efficiently solved using iterative methods. This is typically the case in sparse recovery problems, where a solution can be obtained by solving an optimization problem, but for which the solving procedure is known to be NP-hard [Nat95; TW10]. In such cases, it is possible to introduce simplifying assumptions in order to reduce the computational complexity: e.g. in the aforementioned sparse recovery problem, changing the ℓ_0 constraint to a ℓ_1 constraint makes the problem tractable. An alternative approach consists in using a heuristic algorithm that exploits structural information given by domain knowledge. In this setting, the obtained decision rule is generally suboptimal, as it does not correspond to the optimal inference rule over the full hypothesis class, but it remains computationally tractable [6].

To summarize, the links between model-based methods and the inference rule design framework are explicited in Highlight 3.3.

Highlight 3.3: Model-based methods and inference rule design framework

Model-based methods can be integrated within the inference rule design framework, presented in Highlight 3.2, as follows.

1. **Inference rule type selection.** $\mathcal{H}$ is selected based on domain knowledge. For some problems with sufficient domain knowledge, $\mathcal{H}$ may contain only one function which describes an analytical solution of the problem at hand.

2. **Inference rule tuning.** This tuning step often consists in using iterative algorithms. In the case that domain knowledge is sufficient to derive an analytical solution, i.e. $|\mathcal{H}| = 1$, this step is bypassed.

3. **Tuned inference rule evaluation.** The obtained solution is evaluated using an error measure, which is designed based on the problem at hand.

Limitations

As previously mentioned, the philosophy of model-based methods lies in efficiently leveraging domain knowledge of a specific problem, to design an inference rule that is well adapted to the considered task. In these setups, domain knowledge is often used with

5. An interested reader is referred to [SE23, Section 3.3 pp. 316 - 326] for an extensive presentation of iterative optimizers.

6. This approach is frequently used in wireless communication problems: some examples are presented in [SE23, Section 3.4, pp. 326 - 329].

simplifying assumptions in order to derive an analytical model for the given task, thereby reducing the hypothesis class, or enabling the use of a heuristic algorithm to obtain a solution. This dependency on simplifying assumptions constitutes the main limitation of model-based methods, as presented in Highlight 3.4.

Highlight 3.4: Limitations of model-based methods

Model-based methods often rely on simplifying assumption to derive an analytical model of a given problem. When these assumptions are overly simplistic, or when the system evolves such that the assumptions no longer hold, the resulting inference rule may become suboptimal.

Model-based methods are often characterized by a high bias.

3.1.3 Machine learning methods

The aim of this section is not to provide an in-depth presentation of ML methods, since an entire chapter of this manuscript, Chapter 2, is dedicated to this task. It rather aims at recalling the key concepts of ML methods, see how they integrate within the inference rule design framework, and recall their limitations.

Principles

In direct contrast to model-based methods, the machine learning approach typically assumes minimal or no domain knowledge. Specifically, ML methods typically do not rely on any a priori information regarding an underlying analytical model for the considered task. This stems from the fact that some tasks do not admit an analytical description, or that such a description would present an intractable complexity. For instance, the task of automated question answering, recently popularized by conversational agents such as ChatGPT, Claude or LeChat, does not admit any general analytical model.

By essence, ML methods are data-driven: they rely on the use of data $\mathcal{D}$ to select an inference rule f mapping a context to a decision. As previously mentioned, this constitutes the major difference with model-based methods. To design an inference rule, rather than relying on a priori information, often combined with simplifying assumptions, the ML approach directly leverages available data of the problem, e.g. context/decision pairs in a supervised setting. This data dependency provides greater flexibility in the selected rule: unlike model-based methods, deviations from simplifying problem assumptions are implicitly accounted for during the tuning process on non-ideal data. From a formal perspective, selecting an inference rule using the ML paradigm amounts to solving an

empirical risk minimization problem, described in Eq. (2.13) and recalled below:

$$\begin{aligned} \underset{f_{\boldsymbol{\theta}}}{\text{minimize}} \quad & \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \mu_{\mathcal{Y}}\left(f_{\boldsymbol{\theta}}\left(x\right), y\right), \\ \text{subject to} \quad & f_{\boldsymbol{\theta}} \in \mathcal{H} \end{aligned} \tag{3.15}$$

where $f_{\boldsymbol{\theta}}$ denotes the inference rule, parameterized by $\boldsymbol{\theta}$, (x, y) a context/decision pair, $\mathcal{D}$ the data, $\mu_{\mathcal{Y}} \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ an error measure, and $\mathcal{H}$ the hypothesis class. In this context, the data $\mathcal{D}$ is leveraged to tune the parameters $\boldsymbol{\theta}$. Note that, as for model-based methods, the optimization problem described in Eq. (3.15) may admit an analytical solution, but is most often tackled using iterative solvers. This is particularly true with deep learning methods [7], where the complexity of the composed candidate rules typically precludes closed-form solutions.

Remark 3.2. *Domain knowledge may be used to restrict the hypothesis class to a set of methods well-suited to the studied problem, e.g. using CNNs for image recognition tasks, or to design the error measure and potential regularizers, e.g. using the cross-entropy loss for classification tasks. In fact, using this domain knowledge [8] to design the learning method can be seen as a form of model-based machine learning, and will be made explicit in Section 3.2.*

In summary, the links between machine learning methods and the inference rule design framework are explicited in Highlight 3.5.

Highlight 3.5: Machine learning methods and inference rule design framework

Machine learning methods can be integrated within the inference rule design framework, presented in Highlight 3.2, as follows.

1. **Inference rule type selection.** $\mathcal{H}$ is selected based on the problem at hand.
2. **Inference rule tuning.** This tuning step consists of the training phase of a ML method, i.e. selecting a mapping $f_{\boldsymbol{\theta}} \in \mathcal{H}$ based on the available data.
3. **Tuned inference rule evaluation.** The tuned ML-method is evaluated using an error measure, allowing to evaluate the generalization error.

7. See Section 2.4.
8. Known as the *inductive bias* in the ML field.

Limitations

The data dependency of ML methods gives rise to several concerns [9], which are listed below and summarized in Highlight 3.6.

- **Data.** At the end of the tuning process on data $\mathcal{D}$, the selected rule is expected to present satisfactory performance over the entire context space $\mathcal{X}$. As the tuning process rely on empirical risk minimization, this is attained only if the data accurately represents the context/decision dynamics, i.e. if the estimated joint distribution $\hat{p}_{x,y|\mathcal{D}}$ on $\mathcal{D}$ is close to the actual problem distribution $p_{x,y}$. This typically give rises to the well-known generalization issue: the tuned rule presents satisfactory results on the data used during training, but performs poorly on new data. Several approaches can be envisioned to mitigate this issue, including the use of efficient tuning strategies, or the incorporation of additional data.

- **Complexity.** Modern ML methods, especially when considering deep learning, rely on highly parameterized mappings, sometimes involving billions of parameters as in large language models. Tuning such models is extremely complex and incurs substantial time costs. Moreover, the computational complexity of these methods can be prohibitively high, which limits their usability in resource-constrained settings such as embedded systems.

- **Interpretability.** Owing to their compositional structure, deep learning methods are often regarded as black boxes. Typically, after training, a neural network can achieve excellent performance on a given task. However, explaining why a specific input leads to a particular inferred decision is generally very difficult.

Highlight 3.6: Limitations of machine learning methods

ML methods rely on limited domain knowledge and primarily exploit data to select an inference rule. To achieve satisfactory performance on complex problems, these methods often require highly parameterized mappings, which demand extensive training time, large amounts of data, and generally offer limited interpretability.

ML methods, especially neural networks, are generally difficult to train and often lack interpretability.

9. Note that a detailed explanation of the deep learning paradigm limitations is proposed in Section 2.4.

Table 3.1: Summary of the differences between the MB, ML and MB-ML approaches in the context of the inference rule design framework.

Design steps	MB	ML	MB-ML
Selection	Domain knowledge	Generic models	Domain knowledge
Tuning	Iterative or direct solution	Training on data	Refining a solution on data
Evaluation	Problem-specific error measure		

3.2 Towards model-based machine learning

The model-based machine learning paradigm [MLE21; Shl+21; SEB22; Shl+23; SE23] originates from the limitations of both purely model-based and purely data-driven methods. On the one hand, model-based methods are computationally efficient and provide physically interpretable results, but they are often affected by a strong bias due to their reliance on simplifying assumptions. On the other hand, machine learning methods can directly adapt from data, without requiring an analytical model of the problem at hand, but they often entail significant computational complexity and provide limited interpretability. The model-based machine learning paradigm seeks to combine the strengths of both paradigms, while mitigating their respective weaknesses. A general definition of this paradigm is proposed in Definition 3.2.

Definition 3.2: MB-ML method

A data processing method lies under the realm of model-based machine learning if it leverages domain knowledge about a specific problem, to **structure**, **optimize**, or **initialize** a learning algorithm designed to solve that problem.

Note that the three proposed categories are not mutually exclusive: e.g. a MB-ML method may simultaneously leverage domain knowledge to structure and optimize a learning algorithm. A detailed explanation of each category, along with illustrative examples, is provided in the following paragraphs. Additionally, Table 3.1 presents how the MB-ML approaches integrate with the inference rule design paradigm, alongside the MB and ML approaches.

3.2.1 Structure

This category encompasses every approach that uses domain knowledge to propose a task-adapted structure for the learning method. From a formal perspective it amounts to using domain knowledge to restrict the hypothesis class to a subset that is more representative of the studied problem: an illustration of this concept is provided in Fig. 3.1. In the ML

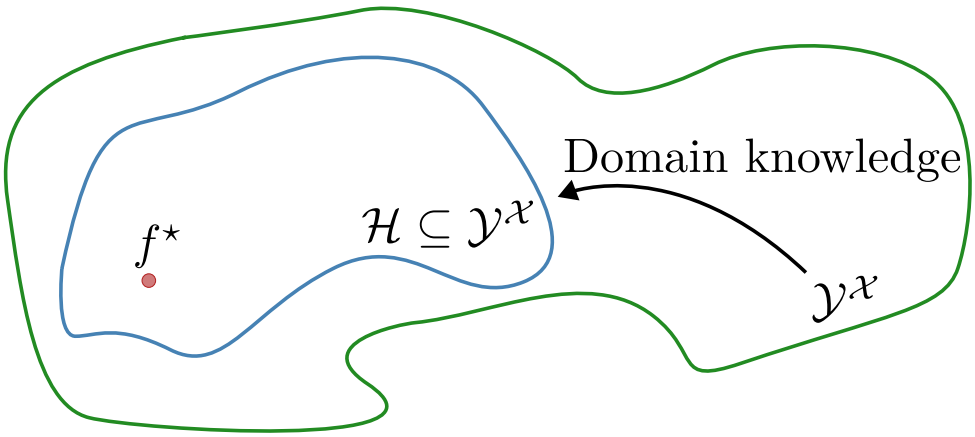


Figure 3.1: Illustration of the use of domain knowledge to structure ML methods. In this setting, domain knowledge is leveraged to restrict the hypothesis class $\mathcal{H}$ to a subset of $\mathcal{Y}^\mathcal{X}$, i.e. the set of candidate mappings, with the hope that it is sufficiently informative so that the optimal mapping $f^\star$ lies within $\mathcal{H}$.

field, this principle is referred to as introducing an inductive bias. Structuring learning methods can be achieved using different approaches: selecting a type of ML methods that are well-adapted to a given problem, or enhancing a model-based approach by replacing specific components with learnable counterparts. It is proposed to illustrate this concept through two specific approaches: neural network-aided (NN-aided) inference [10] and deep unfolding [11].

Remark 3.3. *While the explicit conceptualization of the model-based machine learning paradigm is relatively recent* [12]*, an illustration of the use of its underlying principles can be traced back to the introduction of CNNs* [13]*. Indeed, CNNs were initially proposed for image recognition tasks under the key assumption that translating an image does not alter its semantic content. This prior knowledge was directly embedded into the neural architecture design through the use of convolutional layers. In this sense, CNNs represent an early instance of the model-based paradigm: the a priori knowledge that semantic content is translation-invariant served to restrict the hypothesis class to a family of neural networks well-suited to the problem.*

NN-aided inference

NN-aided inference leverages the adaptability of neural networks to replace computational components of model-based methods that rely on overly simplifying assumptions. This approach preserves the overall structure of the inference rule, which remains grounded in

10. An interested reader is referred to [SE23, Section 5.5, pp. 383 - 396].
11. An interested reader is referred to [SE23, Section 5.3, pp. 368 - 375] or [MLE21].
12. Seminal papers on that particular topic originates from the early 2020s.
13. See Section 2.4.2.

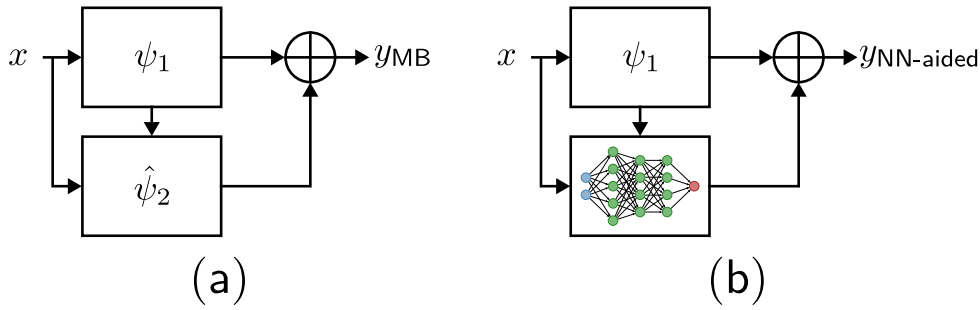


Figure 3.2: Illustration of the NN-aided inference approach on the example presented in Eq. (3.16). (a) denotes the model-based approach, where domain knowledge is leveraged to exactly model ψ_1 and approximately model ψ_2 as $\hat{\psi}_2$. (b) denotes the NN-aided inference approach, where the estimated $\hat{\psi}_2$ is replaced by a generic neural network.

domain knowledge, while introducing data-driven approaches to learn components that are affected by missing of incomplete a priori knowledge.

In order to better grasp this concept, it is proposed to consider a problem whose optimal inference rule can be expressed as:

$$f^\star \colon x \to \psi_1(x) + \psi_2(x, \psi_1(x)), \tag{3.16}$$

where $\psi_1 \colon \mathcal{X} \to \mathcal{Y}$, and $\psi_2 \colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$, are two problem-dependent functions. It is assumed that domain knowledge provides an exact characterization of ψ_1, but relies on simplifying assumptions to model a surrogate $\hat{\psi}_2$ for ψ_2. In the model-based setting, the resulting inference rule takes the form:

$$f_{\mathsf{MB}} \colon x \to \psi_1(x) + \hat{\psi}_2(x, \psi_1(x)). \tag{3.17}$$

On can remark that the imperfect domain knowledge, i.e. surrogate model $\hat{\psi}_2$, translates to suboptimal performance [14]. In contrast, the NN-aided inference approach consists in replacing the approximated $\hat{\psi}_2$ with a model agnostic neural network, $f_{\boldsymbol{\theta}}$, and leverages available data to train it to recover ψ_2. The resulting rule takes the following form:

$$f_{\mathsf{NN\text{-}aided}} \colon x \to \psi_1(x) + f_{\boldsymbol{\theta}}(x, \psi_1(x)). \tag{3.18}$$

This approach preserves the structural information originating from domain knowledge, allowing interpretability and computational efficiency, while exploiting the adaptability of neural networks to compensate for inaccurate domain knowledge. The comparison of both approaches is illustrated in Fig. 3.2.

The NN-aided inference approach has been used for several tasks such as dynamic system tracking [Rev+21; Rev+22], direction of arrival estimation [Shm+23; Shm+25; Cha+25c], or integrated sensing and communication system design [MatRam+23; Ma-

14. From a risk perspective, $\hat{\psi}_2 \neq \psi_2$ yields $\mathsf{R}(f_{\mathsf{MB}}) > \mathsf{R}(f^\star)$.

tRam+24; MatRam+25a].

Highlight 3.7: NN-aided inference and differentiability

While NN-aided inference improves performance and preserves interpretability, it comes at the cost of training an additional neural network. In practice, this training is typically performed using first-order optimization methods such as gradient descent, which imposes the overall inference rule to be differentiable.

Deep unfolding

The deep unfolding methodology, first introduced in [GL10], consists in *unfolding* iterative algorithms into neural networks, where each iteration of the algorithm is mapped to a layer that can be independently trained. This approach can be seen as a specific instance of the NN-aided methodology, in which the model-based approach does not yield an exact analytical solution, but instead leads to an optimization problem, solvable through an iterative solver.

It is proposed to illustrate this concept through a generic problem, whose model-based solution can be expressed as:

$$f_{\mathsf{MB}} \colon x \to \psi\left(y^{(K)}\right), \tag{3.19}$$

where K denotes an iteration number, and ψ denotes a problem-specific iterative mapping, such that:

$$\begin{cases} y^{(k)} = \psi\left(y^{(k-1)}\right), k \in \mathbb{N}^* \\ y^{(0)} = x \end{cases}. \tag{3.20}$$

In this setting, domain knowledge was leveraged to derive an inference rule expressed as an optimization problem, whose solution can be approximated using an iterative solver, as presented in Eq. (3.19) and Eq. (3.20). In practice, domain knowledge often guides the choice of certain parameters of the iterative solver, i.e. it influences ψ in Eq. (3.19). As a result, the solution obtained through Eq. (3.19) relies heavily on the domain knowledge accuracy. The deep unfolding methods consist on defining a neural network $f_{\boldsymbol{\theta}}$ such that:

$$\psi\left(y^{(K)}\right) \simeq \left(f_{\boldsymbol{\theta}_K} \circ f_{\boldsymbol{\theta}_{K-1}} \circ \cdots \circ f_{\boldsymbol{\theta}_1}\right)(x) \tag{3.21}$$

$$= f_{\boldsymbol{\theta}}(x), \tag{3.22}$$

which amounts to considering each iteration of the solver as a layer of a neural network: this is illustrated in Fig. 3.3.

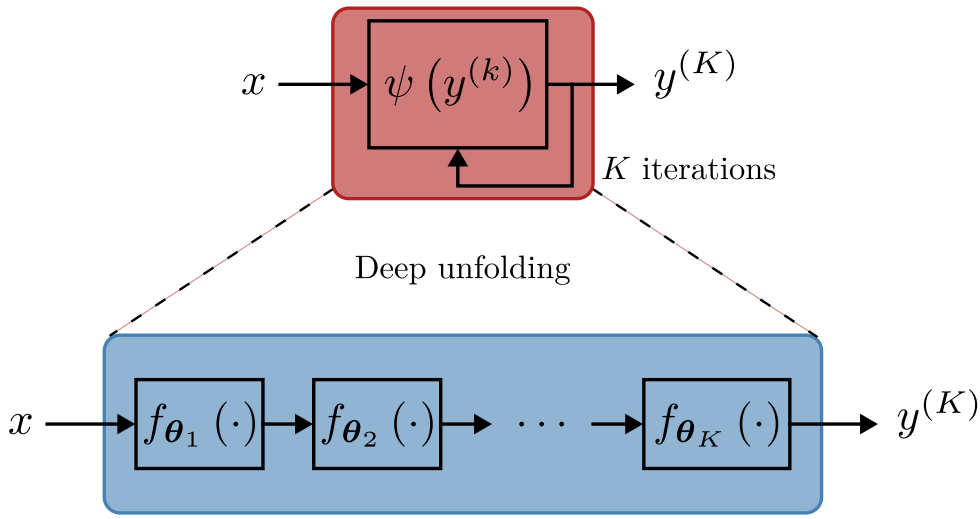


Figure 3.3: Illustration of the deep unfolding method: an iterative solver is unfolded as a neural network $f_{\boldsymbol{\theta}}$, that is trained on available data (based on Fig. 5.6 in [SE23]).

Highlight 3.8: Deep unfolding strategies

The neural network structure obtained by unfolding an iterative optimizer can be exploited in different ways, including:

- Preserving the overall structure of the iterative algorithm, while allowing the hyperparameters to be optimized independently at each layer.
- Using the unfolded structure as an initialization, and subsequently refine each layer as in a traditional neural network.

The deep unfolding method has been successfully applied to several problems ranging from sparse recovery [GL10; YLM22; CLMR23], to non-negative matrix factorization [HRW14], or channel decoding [BSS19], and constitutes the core of Chapter 6.

3.2.2 Optimize

From a general perspective, this category encompasses all methods that incorporate domain knowledge into the optimization steps of a learning algorithm. This broad definition includes a wide range of methods such as the introduction of physics-inspired regularizers, the learning of hyperparameters, or the incorporation of physics-based optimization constraints. It is proposed to illustrate this category with the learned optimizers and PINNs methods.

Learned optimizers

In many problems, domain knowledge can be leveraged to derive an inference rule formulated as a parameterized mapping. The quality of the resulting solution then depends on the tuning of these parameters, whether manually specified based on prior knowledge, adjusted through hand-crafted heuristics, or set via iterative solvers, which themselves introduce additional parameters. The learned optimizer approach [15] leverages available data and efficient optimization techniques, such as gradient descent, to automatically tune the parameters of the inference rule, and those of the tuning procedure itself.

From a formal perspective, given a certain problem whose optimal inference rule can expressed as:

$$f^{\star}\colon x \to \psi\left(x\right), \tag{3.23}$$

it is assumed that domain knowledge can be used to derive an inference rule that writes as:

$$f_{\mathsf{MB}}\colon x \to f_{\boldsymbol{\theta}}\left(x|\boldsymbol{\zeta}\right), \tag{3.24}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$ are known as the *objective parameters* and *hyperparameters*. Objective parameters correspond to the learnable parameters of the model, which are optimized during training to fit $f_{\boldsymbol{\theta}}$ to the true underlying function ψ. In contrast, hyperparameters denotes the parameters of the learning process itself, such as those defining the behavior of iterative solvers. For instance, in the context of neural networks, objective parameters consists in the neural architecture learnable parameters, while hyperparameters typically include the learning rate, epoch number, batch size, temperature, or dropout rate [16]. The learned optimizer approach consists in using available data $\mathcal{D}$, to learn both $\boldsymbol{\theta}^{\star}$ and $\boldsymbol{\zeta}^{\star}$, yielding the optimized inference rule:

$$f_{\mathsf{MB\text{-}ML}}\colon x \to f_{\boldsymbol{\theta}^{\star}}\left(x|\boldsymbol{\zeta}^{\star}\right). \tag{3.25}$$

As an example, assuming a decision space $\mathcal{Y}$, error measures $\mu^{\boldsymbol{\theta}}\colon \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}^{+}$ and $\mu^{\boldsymbol{\zeta}}\colon \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}^{+}$, the updated parameters at step k, using the gradient descent optimization method, are:

$$\begin{cases} \boldsymbol{\theta}^{(k)} \leftarrow \boldsymbol{\theta}^{(k-1)} - \lambda\nabla_{\boldsymbol{\theta}}\mu^{\boldsymbol{\theta}}\left(f_{\boldsymbol{\theta}}\left(x|\boldsymbol{\zeta}\right), y\right)|_{\boldsymbol{\theta}^{(k-1)},\boldsymbol{\zeta}^{(k-1)}} \\ \boldsymbol{\zeta}^{(k)} \leftarrow \boldsymbol{\zeta}^{(k-1)} - \lambda\nabla_{\boldsymbol{\zeta}}\mu^{\boldsymbol{\zeta}}\left(f_{\boldsymbol{\theta}}\left(x|\boldsymbol{\zeta}\right), y\right)|_{\boldsymbol{\theta}^{(k-1)},\boldsymbol{\zeta}^{(k-1)}} \end{cases}, \tag{3.26}$$

where $(x, y) \in \mathcal{D}$, and $\lambda \in \mathbb{R}^{+}$ is the learning rate. Fig. 3.4 illustrates this approach, which is often applied in practice: e.g., [Kla+25] combines deep unfolding and learned optimization for channel estimation and precoding in hybrid MIMO systems.

Remark 3.4. *The main distinction between the learned optimizer and NN-aided inference approaches is that, in the learned optimizer, the overall structure provided by the model-based approach is retained while some of its components, such as some hyperparameters,*

15. An interested reader is referred to [SE23, Section 5.2, pp. 359-368] for an extensive presentation with examples.

16. See [SE23, Section 5.2, pp. 359 - 360] for a detailed distinction between the two types of parameters.

are learned, whereas the NN-aided inference approach consists in replacing certain parts of the structure with neural networks.

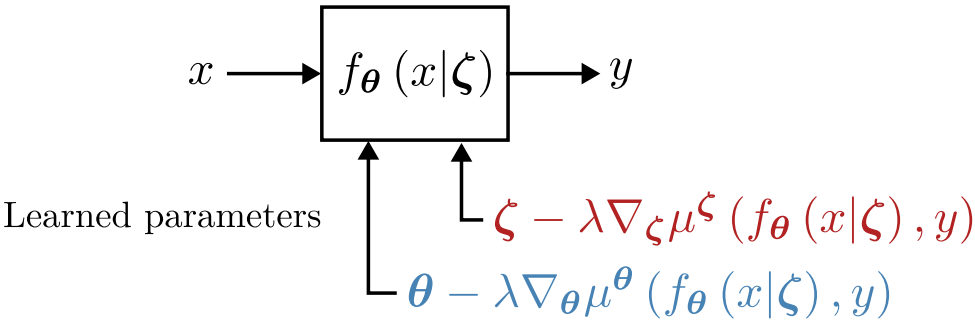


Figure 3.4: Illustration of the learned optimizer approach: available data and a learning method (here gradient descent) are leveraged to optimize the parameters $\boldsymbol{\theta}$, $\boldsymbol{\zeta}$ of an inference rule whose structure is based on domain knowledge.

Highlight 3.9: Learned optimizers as rapidly tuned inference rules

Owing to the efficiency of modern deep learning frameworks, the learned optimizer approach constitutes a promising alternative to traditional hand-crafted tuning strategies, which are typically time-consuming and often require a priori knowledge to restrict the parameter search spaces. However, since most modern learning techniques rely on first-order optimization, the learned optimizer methodology requires the designed inference rule to be differentiable with respect to its parameters.

Physics-Informed Neural Networks

PINNs [RPK19] represent a specific class of neural networks that incorporate domain knowledge in the optimization process. Specifically, PINNs introduce regularizers derived from partial differential equations (PDEs) in the optimization objective.

It is proposed to illustrate this approach on the well-known 2D wave equation, whose solution $u(p_x, p_y, t)$ satisfies the following PDE:

$$\frac{\partial^2}{\partial t^2} u(p_x, p_y, t) - c^2 \left(\frac{\partial^2}{\partial p_x^2} u(p_x, p_y, t) + \frac{\partial^2}{\partial p_y^2} u(p_x, p_y, t) \right) = 0, \tag{3.27}$$

where p_x, p_y, and t respectively represent locations and time, and c is the propagation speed of the wave in the considered medium. This PDE can be solved by finding a neural

network $f_{\boldsymbol{\theta}}$ that approximates $u(p_x, p_y, t)$. Formally, denoting $\mathcal{Y} = \mathrm{dom}^{\mathsf{c}}(u(p_x, p_y, t))$, and given a dataset $\mathcal{D}$ of positions, times, and associated wave field, the optimization problem can be expressed as:

$$\begin{aligned} \underset{f_{\boldsymbol{\theta}}}{\text{minimize}} \quad & \frac{1}{|\mathcal{D}|} \sum_{((p_x,p_y,t),y)\in\mathcal{D}} \mu\left(f_{\boldsymbol{\theta}}(p_x, p_y, t), y\right) + \psi\left(f_{\boldsymbol{\theta}}, p_x, p_y, t\right), \\ \text{subject to} \quad & f_{\boldsymbol{\theta}} \in \mathcal{H} \end{aligned} \tag{3.28}$$

where $\mathcal{H}$ denotes the hypothesis class, $\mu\colon \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}^+$ an error measure, and ψ a regularizer. The distinguishing feature of the PINN paradigm lies in the design of the regularizer. Specifically, the regularizer is said to be physics-based, as it is derived from the underlying PDE of the problem, i.e.:

$$\psi\left(f_{\boldsymbol{\theta}}, p_x, p_y, t\right) = \left| \frac{\partial^2}{\partial t^2} f_{\boldsymbol{\theta}}(p_x, p_y, t) - c^2 \left(\frac{\partial^2}{\partial p_x^2} f_{\boldsymbol{\theta}}(p_x, p_y, t) + \frac{\partial^2}{\partial p_y^2} f_{\boldsymbol{\theta}}(p_x, p_y, t) \right) \right|. \tag{3.29}$$

The optimization objective thus accounts for:

1. A data-driven error term, measuring how accurately the learned network models the given data: $\mu\left(f_{\boldsymbol{\theta}}(p_x, p_y, t), y\right)$.

2. A physics-based error term, quantifying the extent to which the learned network respects the underlying physical principles: $\psi\left(f_{\boldsymbol{\theta}}, p_x, p_y, t\right)$.

Note that it is also possible to add another objective term that takes into account boundary conditions of the considered problem. An illustration of the PINN paradigm applied to the wave equation is proposed in Fig. 3.5.

PINNs have been successfully used on a wide class of tasks, especially in physics, where PDEs are often available to describe the problems at hand. Indeed, PINNs provide satisfying solutions to problems ranging from fluid dynamics [Lu+21], to electromagnetic metamaterial design [FZ20]. An interested reader is referred to [Cuo+22] for a comprehensive review of problems on which PINNs can be applied.

Highlight 3.10: PINNs strengths

PINNs enable the training of neural architectures in a physics-informed manner, ensuring that the resulting solution satisfies the physical laws describing the problem at hand. Owing to the ubiquity of partial differential equations across several scientific fields, and the automatic differentiation capabilities of modern deep learning frameworks, PINNs have emerged as a promising paradigm for many applications.

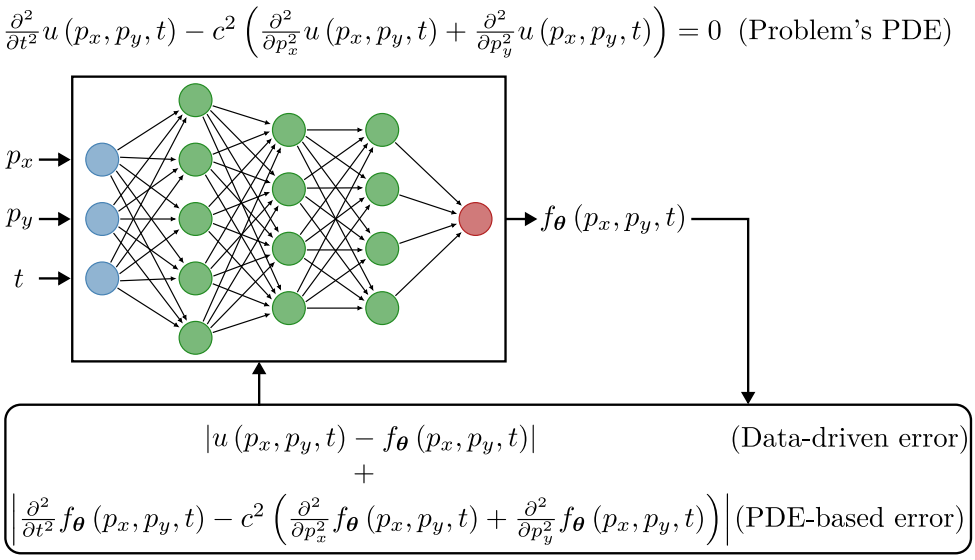


Figure 3.5: Illustration of the PINNs paradigm: a neural network is trained to minimize a combination of a data-driven loss and a physics-based regularization term derived from the underlying PDE of the problem.

Remark 3.5. *Note that, although PINNs have demonstrated promising results across various applications, several challenges remain concerning their training. In particular, studies have shown that the data and physics loss terms may pursue competing objectives, often resulting in slow convergence. This interplay effectively turns the training process into a multi-objective optimization problem, where determining appropriate weights for each objective constitutes a challenging task. An interested reader is referred to [Roh+23] for a detailed discussion.*

3.2.3 Initialize

This approach is closely related to the previous optimization approach. It encompasses all methods that uses domain knowledge to provide an initialization to a learning algorithm. Taking the example presented in Eq. (3.24), the initialization approach consists in leveraging domain knowledge to initialize $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$, that are subsequently optimized using a learning method. For instance, in [YLM22; CLMR23], this initialization approach is applied to a sparse recovery problem: a dictionary is initialized using a physical propagation channel model, and is subsequently optimized through gradient descent. An illustration of this approach is provided in Fig. 3.6.

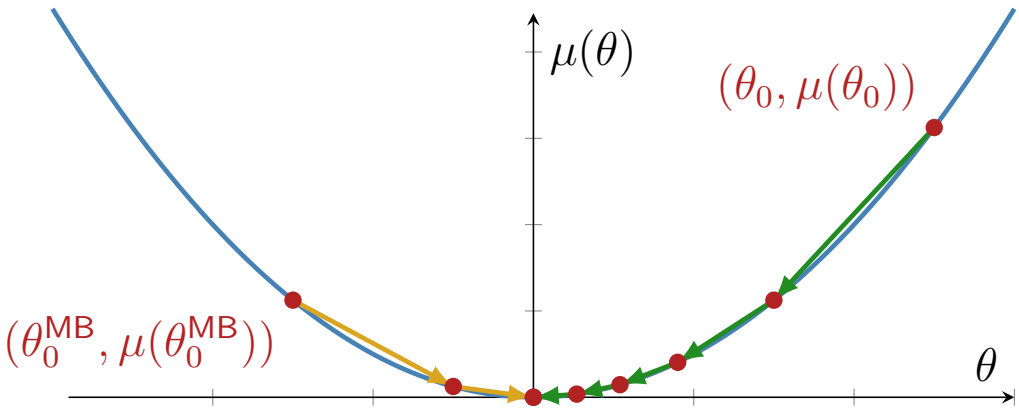


Figure 3.6: Illustration of the model-based initialization approach in a one-dimensional setting: $\mu(\theta)$ denotes the error measure wrt. the learnable parameter θ, θ_0 denotes a model-agnostic initialization, and θ_0^{MB} denotes the model-based initialization. It can be seen that the model-based initialization enables faster convergence.

Highlight 3.11: Initializing yields faster convergence

Initializing the parameters of a learning method based on domain knowledge generally enables faster convergence towards a satisfactory solution. From an error measure perspective, this initialization scheme generally provides a favorable starting point in the parameter space, allowing the optimization procedure to reach a satisfactory minimum in a few iterations.

3.3 Application to the wireless channel

This section aims at providing a concise overview of how the model-based machine learning paradigm is leveraged across the contributions forming the core of this manuscript. The goal is not to present each contribution in full detail, but to highlight how the proposed methods are connected to the MB-ML framework. As stated in the previous sections, the MB-ML paradigm is based on three key ingredients, summarized below:

- **Problem.** A mathematical formulation of the studied question, typically expressed as an optimization problem.
- **Learning algorithm.** A method that leverages available data to provide a solution to the problem at hand.
- **Domain knowledge.** All a priori information about the studied problem, including physics-based simplifying assumptions or analytical models.

Decades of research in wireless communications have produced an extensive collection of analytical models for many classical problems, making the MB-ML approach particularly well suited to this field [17]. As noted above, all contributions of this thesis follow the MB-ML paradigm: although the problems, learning methods, and the extent of domain knowledge may vary, they all build upon the same simplifying assumption, presented below.

Working hypothesis

(H1) Given an emitter and a receiver, the propagating waves between them are assumed to undergo a phase shift and attenuation that are both proportional to the propagation distance.

In each of the proposed contributions, (H1) can be used to:

- **Formulate a propagation channel model**: Chapters 4, 5, 6 and 7.
- **Advocate for the use of a specific distance**: Chapters 8, and 9.

Additionally, each contribution addresses a distinct wireless communication problem that can be categorized within the research axes (A1)-(A3). A detailed presentation of each problem is presented in the following subsections.

3.3.1 Radio environment compression and localization

The contributions related to research axis (A1) are presented in Chapter 4 and 5. These chapters present how to perform radio environment compression using a model-based neural network, and how to use this network to solve a localization task.

Radio environment compression: Chapter 4

The core of this problem lies in selecting an inference rule for the location-to-channel mapping, i.e. finding a mapping that infers propagation channel coefficients $\mathbf{H}$, directly from a location $\mathbf{x}$. Selecting an inference rule that approximates this mapping can be interpreted as a radio environment compression task, since the rule encodes the entire radio environment by mapping low-dimensional locations to their corresponding high-dimensional channel coefficients.

While physics-based propagation models can provide approximated analytical solutions to this problem, they often rely on simplifying assumptions, and involve parameters that require extensive tuning, which limits their practical applicability. The contribution in Chapter 4 consists in structuring a neural network $f_{\boldsymbol{\theta}}$, based on a physical propagation channel model that relies on (H1), to learn the location-to-channel mapping.

17. This is not the case in several other fields. For example, due to the inherent complexity of the considered problems, fields such as image and video processing often lack analytical models.

Problem 3.1: MB-ML for radio environment compression

Formally, the location-to-channel mapping writes as:

$$\begin{aligned} f : \mathcal{X} &\longrightarrow \mathcal{Y} \\ \mathbf{x} &\longrightarrow \mathbf{H}(\mathbf{x}), \end{aligned} \tag{3.30}$$

where $\mathcal{X}$ defines the location space, i.e. $\mathcal{X} \subseteq \mathbb{R}^3$, and $\mathcal{Y}$ defines the channel coefficients space, i.e. $\mathcal{Y} \subseteq \mathbb{C}^{m \times n}$ with specific channel dimensions m and n.

Approximating this mapping can be formally defined as solving the following optimization problem:

$$\begin{aligned} \underset{f_{\boldsymbol{\theta}}}{\text{minimize}} \quad & \mathbb{E}_{(\mathbf{x},\mathbf{H}(\mathbf{x}))\sim p_{\mathbf{x},\mathbf{H}}} \left[\| f_{\boldsymbol{\theta}}(\mathbf{x}) - \mathbf{H}(\mathbf{x}) \|_{\mathsf{F}}^2 \right]. \\ \text{subject to} \quad & f_{\boldsymbol{\theta}} \in \mathcal{H} \end{aligned} \tag{3.31}$$

The proposed MB-ML approach in Chapter 4 consists in using (H1) to define a neural network whose structure is well-adapted to this specific learning task.

Wireless localization: Chapter 5

In wireless communication, localization, also referred to as positioning, consists in determining the position of a UE, based on given measurements. Multiple model-based methods have been proposed for this task, some of which rely on information collected from multiple base stations [GG05]. An alternative approach, known as fingerprinting [18], leverages a database of location/channel coefficient pairs. Given a channel matrix, the estimated position is obtained as the location corresponding to the most similar channel matrix in the dictionary [19]. The accuracy of fingerprinting methods is therefore highly dependent on the spatial density of the used database, and on the level of distortion in the channel coefficients. For example, a synthetic database generated with a propagation channel model often yields poor performance in real environments due to the simplifying assumptions inherent to the channel model.

The contribution presented in Chapter 5 builds on the previously introduced neural network that learns the location-to-channel mapping. This neural network, structured using (H1), is used to enhance the accuracy of a fingerprinting-based localization method. Indeed, since this neural network encodes a continuous mapping between locations and channel coefficients, it can be exploited to augment a fingerprinting database with additional realistic entries, thereby increasing its spatial density. Furthermore, a model-based

18. An interested reader is referred to [Nes+20] for an extensive survey.
19. The notion of channel similarity is deliberately kept vague at this stage: a formal definition will be provided in Chapter 5.

optimization of the proposed localization method is introduced, relying on a propagation channel model derived from (H1).

Problem 3.2: MB-ML for wireless localization

Formally, the localization problem based on channel coefficients involves finding a mapping from the channel space $\mathcal{X} \subseteq \mathbb{C}^{m \times n}$, with specific channel dimensions m and n, to the location space $\mathcal{Y} \subseteq \mathbb{R}^3$:

$$\begin{aligned} f \colon \mathcal{X} &\longrightarrow \mathcal{Y} \\ \mathbf{H}(\mathbf{x}) &\longrightarrow \mathbf{x}. \end{aligned} \tag{3.32}$$

Approximating this mapping can be written as the following optimization problem:

$$\begin{aligned} \underset{f_{\boldsymbol{\theta}}}{\text{minimize}} \quad & \mathbb{E}_{(\mathbf{H}(\mathbf{x}),\mathbf{x}) \sim p_{\mathbf{H},\mathbf{x}}} \left[\| f_{\boldsymbol{\theta}}(\mathbf{H}(\mathbf{x})) - \mathbf{x} \|_2^2 \right]. \\ \text{subject to} \quad & f_{\boldsymbol{\theta}} \in \mathcal{H} \end{aligned} \tag{3.33}$$

The traditional fingerprinting method consists of the solution:

$$f_{\boldsymbol{\theta}}(\mathbf{H}(\mathbf{x})) = \underset{\tilde{\mathbf{x}} \in \mathcal{G}}{\arg\max}\, \text{sim}(\mathbf{H}(\mathbf{x}), \mathbf{H}(\tilde{\mathbf{x}})), \tag{3.34}$$

where $\mathcal{G}$ denotes a database, and $\text{sim}(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ is a channel similarity measure. Since the accuracy of fingerprinting-based localization methods is constrained by the spatial density of the database $\mathcal{G}$, it is proposed to enhance this approach through MB-ML. Specifically, the model-based neural network introduced in Chapter 4 is used to infer channel coefficients at arbitrary locations, thereby increasing the spatial density of the locations in the database.

3.3.2 Joint parametric and hardware impairments estimation

The contributions related to research axis (A2), presented in Chapters 6 and 7, address a joint estimation problem. In both cases, the proposed methods aim to estimate task-specific parameters alongside certain hardware impairments (HWIs) related to the considered system. A common observation across these setups is that the performance of the parametric estimation task is significantly affected by the lack of HWI knowledge. To address this challenge, the proposed contributions build upon (H1) to develop an analytical model that addresses the estimation task, which is subsequently refined through the application of learning algorithms to learn the unknown HWIs, while simultaneously estimating the parameters of interest. This can be formally defined as follows.

Problem 3.3: MB-ML for joint parametric and HWI estimation

Given an observation $\mathbf{x} \in \mathcal{X}$, and unknown parameters $\mathbf{y} \in \mathcal{Y}$, the aim of parametric estimation is to recover $\mathbf{y}$ from $\mathbf{x}$, which amounts to estimate the mapping:

$$\begin{aligned} f_{\boldsymbol{\zeta}} \colon \mathcal{X} &\longrightarrow \mathcal{Y} \\ \mathbf{x} &\longrightarrow \mathbf{y}, \end{aligned} \tag{3.35}$$

where $\boldsymbol{\zeta}$ denotes HWIs associated with the system at hand, which are themselves to be estimated. This can be rewritten as the following optimization problem:

$$\begin{aligned} \underset{\tilde{\boldsymbol{\zeta}}}{\text{minimize}} \quad & \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{\mathbf{x},\mathbf{y}}} \left[\mu_{\mathcal{Y}} \left(f_{\tilde{\boldsymbol{\zeta}}} \left(\mathbf{x} \right), \mathbf{y} \right) \right], \\ \text{subject to} \quad & \tilde{\boldsymbol{\zeta}} \in \mathcal{H}_{\boldsymbol{\zeta}} \end{aligned} \tag{3.36}$$

where $\mu_{\mathcal{Y}} \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^{+}$ denotes an error measure characterizing the distortion of the estimated parameters, and $\mathcal{H}_{\boldsymbol{\zeta}}$ the HWIs hypothesis class.

Model-based methods can be used to select an inference rule that solves this inference rule selection problem, based on (H1). However, this approach remains suboptimal as it typically assumes nominal HWIs, and therefore do not optimize $\tilde{\boldsymbol{\zeta}}$. To address this limitation, the MB-ML paradigm is leveraged by enhancing the model-based solution through the integration of a learning method for $\tilde{\boldsymbol{\zeta}}$.

Channel estimation: Chapter 6

Chapter 6 addresses a channel denoising problem, where the objective is to estimate channel coefficients from noisy observations. Leveraging domain knowledge of the problem, based on (H1), the observations can be modeled as a noisy sparse linear combination of frequency response vectors. Building on this model, traditional sparse recovery algorithms can be applied to perform channel denoising. However, these algorithms rely on a dictionary of frequency response vectors, which are highly sensitive to HWIs, such as imperfect frequency offsets knowledge. To address this limitation, the MB-ML paradigm is leveraged, through the deep unfolding approach. Specifically, the well-known matching pursuit (MP) algorithm [MZ93], an iterative sparse recovery method, is unfolded into a neural network. This enables the introduction of learnable parameters, particularly the HWIs, which are learned during training, ultimately leading to improved denoising performance over traditional model-based methods. Table 3.2 summarizes how the generic problem defined in Problem 3.3 specializes to the presented channel denoising task.

Table 3.2: Summary of the used parameters in Chapters 6 and 7, in the context of Problem 3.3. m, n, m', n' and p denote arbitrary problem dimensions.

Contribution	$\mathcal{X}$	$\mathcal{Y}$	$\mu_{\mathcal{Y}}$	ζ
Chapter 6	$\mathbb{C}^{m\times n}$	$\mathbb{C}^{m\times n}$	ℓ_2 norm	Frequency HWIs
Chapter 7	$\mathbb{C}^{m'\times n'}$	$[-\pi/2, \pi/2]^p$	Modified ℓ_2 norm	Antenna HWIs

DoA estimation: Chapter 7

Chapter 7 studies the DoA estimation problem: given measurements from an antenna array, the goal is to estimate the DoAs of impinging signals. Classical model-based methods, such as the multiple signal classification (MUSIC) algorithm, rely on (H1) to represent the received signal as a noisy linear combination of steering vectors. A subspace decomposition technique is then applied to mitigate noise and extract the DoAs. However, as for the channel denoising problem, the elements of the linear combination, in this case the steering vectors, are distorted by antenna HWIs, resulting in degraded DoA estimation performance. To address this issue, the proposed contribution leverages the MB-ML paradigm, in particular the learned optimizer approach, to jointly estimate the DoAs and learn the HWIs. The key idea is to parametrize the steering vectors as a function of the HWIs, and to update the HWI parameters through gradient descent. To enable this, all processing steps require differentiability, which introduces a subsequent contribution: a differentiable extension of the MUSIC algorithm. Table 3.2 presents the different parameters of Problem 3.3, in the context of the DoA estimation problem.

3.3.3 Channel compression through channel charting

The contributions presented in Chapters 8 and 9 consider a task-oriented channel compression problem, which consists of research axis (A3). As discussed in Section 1.3, modern wireless communication schemes rely heavily on channel state information (CSI) to enhance transmission performance. For instance, CSI is used to compute beamforming matrices that direct signals towards the intended users. However, obtaining CSI at the base station entails significant complexity, especially in systems with non-reciprocal channels [20]. Indeed, in these settings, a BS transmits known pilot symbols to UEs, enabling them to estimate the channel coefficients. Then, the estimated CSI must be fed back from the UEs to the BS. Since CSI usually consists of a large number of coefficients, this feedback procedure is costly.

To reduce the complexity of this feedback procedure, a common approach is to compress the estimated CSI into a low-dimensional representation at the UE level, transmit this compressed representation, and subsequently reconstruct the CSI at the BS level. This constitutes the well known CSI compression framework, presented in [WSJ18; Wan+19;

20. Such as frequency division duplex (FDD) systems, where uplink and downlink frequencies differ.

CGS22]. An alternative, task-oriented, approach is to bypass explicit CSI reconstruction, and instead directly infer the quantity of interest, e.g. a precoding matrix or codebook index, from the low-dimensional channel representation. From a formal perspective, this procedure can be expressed as an optimization problem, which is presented below.

Problem 3.4: MB-ML for task-oriented CSI compression

Given channel coefficients $\mathbf{h} \in \mathcal{X}$, the aim of task-oriented CSI compression is to compress $\mathbf{h}$ in a low-dimensional representation $\mathbf{z} \in \mathcal{Z}$, and to decode $\mathbf{z}$ into a quantity of interest $\mathbf{y} \in \mathcal{Y}$, related to the considered task. This framework can be interpreted as estimating an encoding map:

$$\begin{aligned} f\colon \mathcal{X} &\longrightarrow \mathcal{Z} \\ \mathbf{h} &\longrightarrow \mathbf{z}, \end{aligned} \tag{3.37}$$

and a decoding map:

$$\begin{aligned} g\colon \mathcal{Z} &\longrightarrow \mathcal{Y} \\ \mathbf{z} &\longrightarrow \mathbf{y}, \end{aligned} \tag{3.38}$$

with $\dim_{\mathbb{R}}(\mathcal{Z}) \ll \dim_{\mathbb{R}}(\mathcal{X})$. This mapping estimation task translates to the following optimization problem:

$$\begin{aligned} \underset{f_{\boldsymbol{\theta}_\mathbf{e}},\, g_{\boldsymbol{\theta}_\mathbf{d}}}{\text{minimize}} \quad & \mathbb{E}_{(\mathbf{h},\mathbf{y})\sim p_{\mathbf{h},\mathbf{y}}}\left[\mu_{\mathcal{Y}}\left(\left(g_{\boldsymbol{\theta}_\mathbf{d}} \circ f_{\boldsymbol{\theta}_\mathbf{e}}\right)(\mathbf{h}), \mathbf{y}\right)\right], \\ \text{subject to} \quad & \left(f_{\boldsymbol{\theta}_\mathbf{e}}, g_{\boldsymbol{\theta}_\mathbf{d}}\right) \in \mathcal{H}, \\ & \operatorname{dom}^{\mathsf{c}}\left(f_{\boldsymbol{\theta}_\mathbf{e}}\right) = \operatorname{dom}\left(g_{\boldsymbol{\theta}_\mathbf{d}}\right), \\ & \dim_{\mathbb{R}}\left(\operatorname{dom}^{\mathsf{c}}\left(f_{\boldsymbol{\theta}_\mathbf{e}}\right)\right) \ll \dim_{\mathbb{R}}\left(\operatorname{dom}\left(f_{\boldsymbol{\theta}_\mathbf{e}}\right)\right) \end{aligned} \tag{3.39}$$

where $\mu_{\mathcal{Y}}\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ represents a performance measure related to the considered task.

Traditional CSI compression methods typically rely on model-agnostic neural networks for both the encoder $f_{\boldsymbol{\theta}_\mathbf{e}}$ and the decoder $g_{\boldsymbol{\theta}_\mathbf{d}}$, which limits the interpretability of the resulting encoding and decoding schemes. By contrast, the MB-ML paradigm can be used to structure $f_{\boldsymbol{\theta}_\mathbf{e}}$ and $g_{\boldsymbol{\theta}_\mathbf{d}}$ according to (H1), thereby ensuring that the low-dimensional representation space remains interpretable and that the decoding procedure yields satisfactory performance.

In both contributions, the dimensionality constraints presented in Problem 3.4 are satisfied by the encoder/decoder design. Additionally, the MB-ML encoder is obtained through channel charting (CC). CC is an unsupervised machine learning method, that embeds channel coefficients into a low-dimensional pseudo-location space, subject to constraints ensuring that topological structures, specifically spatial neighborhoods, are preserved be-

tween the channel and representation spaces [21].

Efficient beam prediction: Chapter 8

Chapter 8 focuses on a specific wireless system setup known as FDD cell-free massive MIMO systems. In this context, the cell-free structure allows base stations to communicate with each others. The non-heterogeneous nature of cell-free systems [22], and high dimensionality of CSI in massive MIMO (mMIMO) systems, motivate the use of a task-based CSI compression scheme. Specifically, it is proposed to reduce the complexity of the beam-assignment task in the context of discrete beam dictionaries, i.e. codebooks. The proposed approach relies on designing an encoder that embeds CSI into a low-dimensional pseudo-location, and a model-based decoder that maps the pseudo-location to a beam index. This leads to the following procedure, that significantly reduces the overall beam-assignment complexity:

1. A BS sends pilot symbols to a given UE.
2. The UE estimates the CSI and transmits it to the BS.
3. The BS encodes the estimated CSI into a low dimensional pseudo-location.
4. The BS forwards the pseudo-location to all other BSs in the cell-free system.
5. Each BS uses a trained decoder to infer the codebook beam to serve the UE.

In this procedure, domain knowledge, based on (H1), is leveraged to structure both the encoder and decoder as model-based neural networks, yielding superior performance compared to model-agnostic neural networks. A summary of the parameters of Problem 3.4 in the context of this beam-assignment task is provided in Table 3.3.

CSI compression: Chapter 9

Chapter 9 focuses on a traditional FDD mMIMO cellular system. Similarly to the previously discussed cell-free mMIMO setup, precoder design in conventional FDD mMIMO systems is hindered by the high dimensionality of CSI, which induces significant reporting complexity. To address this issue, the MB-ML framework is leveraged to design efficient neural encoders and decoders, thereby reducing the overall complexity of the CSI feedback procedure. Specifically, instead of reconstructing CSI from low-dimensional pseudo-locations for subsequent precoder computations, the proposed method decodes the pseudo-locations directly into a precoder. This is enabled by introducing a beamforming-based error measure into the joint training of the neural encoder and decoder structures. As for the

21. Note that an entire section of Chapter 8, namely Section 8.1, is dedicated to a detailed presentation of the channel charting concept.
22. For instance, base stations may operate on different frequency bands.

Table 3.3: Summary of the used parameters in Chapters 8 and 9, in the context of Problem 3.4. m, n and d denotes arbitrary problem dimensions, with the constraint $d \ll m$.

Contribution	$\mathcal{X}$	$\mathcal{Z}$	$\mathcal{Y}$	$\mu_{\mathcal{Y}}$
Chapter 8	$\mathbb{C}^m$	$\mathbb{R}^d$	$\mathbb{N}^*$	Cross-entropy
Chapter 9	$\mathbb{C}^m$	$\mathbb{R}^d$	$\mathbb{C}^{m\times n}$	Beamforming based

Table 3.4: Summary of the use-cases covered by the different contributions of the thesis.

Use-case	Contribution	MB-ML approach
Estimating	Chapter 4	Structure
	Chapter 5	Structure/Optimize
	Chapter 6	Structure/Initialize
	Chapter 7	Structure/Initialize
Compressing	Chapter 4	Structure
	Chapter 8	Structure/Initialize
	Chapter 9	Structure/Initialize
Controlling	Chapter 8	Structure/Initialize
	Chapter 9	Structure/Initialize

previously presented method, domain knowledge, based on (H1), is incorporated to structure the encoder/decoder architectures. It is shown that the proposed MB-ML method outperforms model-agnostic ML approaches, while substantially reducing the overall CSI reporting complexity. Table 3.3 presents the different parameters of Problem 3.4 in the context of this precoder regression task.

3.3.4 Use-case summary

For each contribution, the associated MB-ML model is applicable to at least one of the following use cases:

- **Estimating**: use the model to estimate parameters of the propagation channel.
- **Compressing**: use the model to compress information associated with the propagation channel.
- **Controlling**: use the model to adapt the considered system parameters to the propagation channel.

Table 3.4 summarizes the use-cases for each contribution and their associated MB-ML approach, namely, structure, optimize, or initialize, as defined in Definition 3.2.

3.4 Conclusion

The goal of this chapter was to introduce the model-based machine learning paradigm, and present how it differs from classical machine learning. Several key aspects can be highlighted, as summarized below.

- Model-based and machine learning methods are embedded within the inference rule design framework, which provides the theoretical foundations for analyzing and comparing these different approaches.
- Model-based methods rely on domain knowledge and simplifying assumptions to derive analytical models describing a given problem. This dependency can introduce high biases, but generally yields solutions with low computational complexity.
- Machine learning methods are data-driven: for a given problem, they learn a solution based on available data, without relying on explicit assumptions. However, they often require a substantial training phase, which incurs significant computational complexity and requires large volumes of data. Furthermore, the performance of trained machine learning models often suffers from limited interpretability, thereby restricting the extent to which their associated inference rules can be understood and trusted.
- The MB-ML paradigm leverages the respective strengths of both approaches: domain knowledge is used to structure, optimize, or initialize learning algorithms, resulting in interpretable solutions with relatively low complexity.
- Owing to the abundance of analytical models, the wireless communication field is particularly well-adapted to the use of MB-ML methods. This thesis focuses on the application of the MB-ML paradigm in three distinct areas: location-to-channel mapping learning, i.e. (A1), hardware impairments compensation, i.e. (A2), and channel compression, i.e. (A3).

Part II

Contributions

Chapter 4

Location-to-channel mapping: learning

This chapter focuses on designing a neural network to learn the location-to-channel mapping, a task made challenging by high-frequency nature of this mapping, and the resulting spectral bias. Section 4.1 introduces the context and formulates the learning problem. Section 4.2 demonstrates how domain knowledge can be leveraged to derive an approximate analytical expression for the considered mapping. Building upon this expression, Section 4.3 proposes a neural architecture specifically adapted to the problem at hand. Finally, Section 4.4 presents experimental results validating the proposed approach, and Section 4.5 present some concluding remarks.

Contents

4.1 Problem setting **112**
- 4.1.1 General introduction 112
- 4.1.2 Related work and contributions 115
- 4.1.3 Problem formulation 116

4.2 Towards a model-based approach **119**
- 4.2.1 Model-based approach 119
- 4.2.2 Discussion: a sparse recovery approach 124

4.3 Model-based neural architecture **125**
- 4.3.1 Main building blocks 125
- 4.3.2 Global architecture 126

4.4 Experiments **128**
- 4.4.1 Learning framework and baselines 128
- 4.4.2 Results and applications 131

4.5 Conclusion **140**

This chapter is mainly based on a journal article published in the *IEEE Journal of Selected Topics in Signal Processing* [Cha+25a], which extends a conference article presented at the *2024 IEEE International Conference on Acoustics, Speech and Signal Processing* [Cha+24b]. Furthermore, the experiments described in this chapter entailed a total computational cost of approximately 255 GPU days.

4.1 Problem setting

4.1.1 General introduction

In the past few years, model-based machine learning have showed great results in channel estimation problems [He+18; WHD21; YLM22; CLMR23], direction of arrival estimation [Shm+25; Cha+25c], channel charting [Yas+22; Yas+23; Yas+24], but also in integrated sensing and communication scenarios [MatRam+24]. It is proposed to further explore this paradigm by examining its capabilities in the context of Problem 3.1, namely radio-environment compression. The central question is whether a model-based neural architecture can be designed to learn the location-to-channel mapping. Since the propagation channel coefficients are inherently related to the user's location, this mapping can, in principle, be captured by a suitably structured neural network. Once trained, such a network would require only the user's location as inputs to generate the corresponding channel coefficients. This approach would be beneficial in many applications: some examples include channel estimation, secure communication mechanisms, resource allocation or interference management, but also radio-environment compression. Indeed, if one achieves near perfect learning of the location-to-channel mapping with the neural network, it could be more efficient to only store the weights of the trained neural network rather than directly storing the channel coefficients.

Learning continuous mappings with neural networks is known as the implicit neural representation (INR) problem. After its great success in the resolution of image processing problems such as novel view synthesis [Mil+21], researchers have started to establish theoretical results on the INRs mapping learning capabilities [RL22; ZRL21; Yuc+22; RML22; SCL24; Sar+24]. While classical architectures, such as MLPs, are universal function approximators [1], it has been shown that they exhibit a bias towards learning low-frequency functions, a phenomenon known as the spectral bias [Rah+19; Cao+21; JohXu+20].

Focus on the spectral bias issue.

The spectral bias phenomenon does not imply that classical MLPs are incapable of approximating functions with high-frequency content [2], but rather that such architectures generally require extended training duration to effectively capture high-frequency features. The theoretical foundation of the spectral bias phenomenon can be explained using the neural tangent kernel theory (NTK), presented in Highlight 4.1. Specifically, it is shown in [Cao+21] that, during gradient descent, a two-layer MLP initially learns components of the target function that lie in the span of the NTK's eigenfunctions associated with the largest eigenvalues [3]. Furthermore, it is also shown that these dominant components correspond to low degree spherical harmonics, i.e. low-frequency content [Cao+21, Sections 3.1, 3.2, pp. 3-4]. An illustration of this phenomenon is presented in Fig. 4.1, where

1. See [Cyb89; HSW89], and Theorem 2.1.
2. This assertion would contradict the universal approximation theorem.
3. Eigenfunctions can be viewed as generalizations of eigenvectors to function spaces.

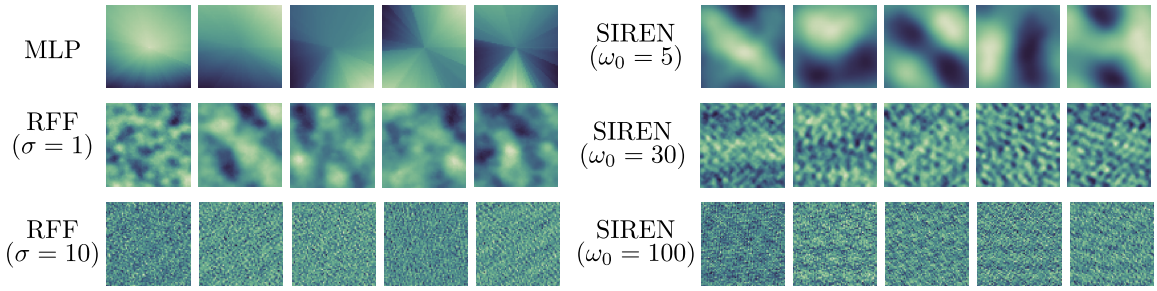


Figure 4.1: Illustration of the first five eigenfunctions of the empirical NTK for different neural networks at initialization. For the RFF and SIREN architectures, the hyperparameters, σ, ω_0, modulate the maximum frequency span in the embedding layer. As the eigenfunctions of the NTK condition the learning performance of neural networks, having eigenfunctions with high frequency content allows to overcome the spectral bias issue. It can clearly be seen that the MLP eigenfunctions present low frequency content, showcasing the spectral bias phenomenon (modified Fig. 5 in [Yuc+22]).

the RFF and SIREN networks are INRs developed to overcome the spectral bias issue, as presented in the subsequent paragraph.

Highlight 4.1: Neural tangent kernel theory

An important theoretical framework for analyzing neural network learning dynamics is the NTK theory [JGH18]. Formally, NTK theory states that training a neural network with sufficiently wide layers using gradient descent in its parameter space is equivalent to performing gradient descent in a reproducing kernel Hilbert space (RKHS). This RKHS is defined by a kernel specific to the considered neural network, known as the NTK. In other words, the training dynamics of a sufficiently wide neural network can be approximated by those of a kernel-based method governed by the NTK.

To address the limitations induced by the spectral bias phenomenon, several specialized architectures have been developed: random Fourier features (RFFs) [RR07; Tan+20], sinusoidal representation networks (SIRENs) [Sit+20], or Gaussian activated radiance fields (GARFs) [Chn+22]. These architectures aim at incorporating high-frequency content into the neural model. The principles underlying these spectral bias resistant designs can be categorized within two approaches: embedding specialization within traditional `ReLU`-MLPs, or embedding replacement.

- **Embedding specialization.** This approach consists on the projection of the input on a higher dimensional space containing high-frequency content, and is commonly referred to as the *positional embedding* method. This approach, firstly introduced

in [Mil+21], aligns with earlier findings in [Rah+19], where the authors showed that introducing a particular high-frequency embedding allowed to ease the learning of high-frequency content in the target mapping. Further work focused in the design of a well adapted embedding layer for the target mapping learning, while the rest of the neural architecture is a traditional `ReLU`-MLP. This approach has been used in [RR07; Mil+21; Tan+20; Pum+21; Par+21; Du+21].

- **Embedding replacement.** Conversely, this approach suggests omitting the initial high-frequency embedding and instead specializing the activation functions of the MLP. This method recently garnered significant attention: in [Sit+20], the authors proposed to replace `ReLU` non-linearities by sine functions, achieving high performance in high-frequency mapping learning. Nevertheless, it has been shown that using sine non-linearities can lead to high sensitivity to the network parameters initialization. To address this, the authors in [Chn+22] proposed replacing sine non-linearities with Gaussian ones, achieving strong performance in image reconstruction problems without requiring intricate network parameters initialization schemes. Additionally, in [RL22], the authors showed that sine non-linearities were part of a broad class of non-linearities that permitted MLPs to learn high-frequency content. They also showed that using non-periodic non-linearities allowed to obtain good performance in novel view synthesis problems. Finally, it has recently been shown in [Sar+23] that complex Gabor wavelets could be used as activation functions, yielding good performance in a wide range of image processing problems.

Theorem 4.1: [Yuc+22] Expressive power of MLPs with sinusoidal embeddings.

Let $f_{\boldsymbol{\theta}} \colon \mathbb{R}^D \to \mathbb{R}$ be a L-layer MLP with polynomial activation functions, i.e. that admits a representation $\sigma(x) = \sum_{k=1}^{K} \alpha_k x^k$. Let us assume that the initial layer of $f_{\boldsymbol{\theta}}$ is a sine embedding layer:

$$\gamma(\mathbf{x}) = \sin(\boldsymbol{\Omega}\mathbf{x} + \boldsymbol{\phi}), \tag{4.1}$$

where $\boldsymbol{\Omega} = [\boldsymbol{\Omega}_1, \cdots, \boldsymbol{\Omega}_T]^\mathsf{T} \in \mathbb{R}^{T \times D}$ and $\boldsymbol{\phi} \in \mathbb{R}^T$ respectively denote the frequency and phase embedding parameters. Then, $f_{\boldsymbol{\theta}}$ can only represent functions of the form:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{\boldsymbol{\omega} \in \mathcal{H}(\boldsymbol{\Omega})} c_{\boldsymbol{\omega}} \sin(\langle \boldsymbol{\omega}, \mathbf{x} \rangle + \phi_{\boldsymbol{\omega}}), \tag{4.2}$$

where:

$$\mathcal{H}(\boldsymbol{\Omega}) \subseteq \left\{ \sum_{t=1}^{T} s_t \boldsymbol{\Omega}_t \middle| s_t \in \mathbb{Z} \text{ s.t. } \sum_{t=1}^{T} |s_t| \leq K^{L-1} \right\}. \tag{4.3}$$

Proof. See Appendix of [Yuc+22]. ■

Theorem 4.1, allows to explain the good learning performance of SIRENs and RFFs for target functions with high frequency content. Indeed, it establishes that MLPs equipped with polynomial activation functions and a sinusoidal embedding layer, are capable of representing functions expressed as linear combinations of certain harmonics of the embedding layer. Consequently, the ability of such architectures to capture high-frequency content is primarily determined by the design of their embedding layer. In particular, [Yuc+22] demonstrates that a simple three-layer SIREN can represent signals composed of infinitely many frequency harmonics, owing to its specialized embedding design.

Location-to-channel mapping and spectral bias

The location-to-channel mapping presents a high-frequency spatial dependence, on the order of the operating wavelength, making its learning a remarkably complex problem due to the aforementioned spectral bias issue. One may then pose the subsequent inquiries: *Are classical INR architectures able to learn the location-to-channel mapping? Is a model-based approach able to learn this mapping? Does the model-based approach outperform INR architectures in terms of learning performance and complexity?*

4.1.2 Related work and contributions

Related work

Learning continuous mappings through neural networks has been extensively studied by the image processing community. For instance, it has been used for image reconstruction problems [Sit+20; Bem+20; CLW21], for 3D scene reconstruction from 2D images [Mil+21; Yar+20; SZW19] but also for dynamic 3D scene reconstruction from 2D images [Pum+21; Par+21; Tre+21; Du+21; Xia+21]. In the signal processing community, machine learning capabilities have been recently leveraged for various problems: e.g. ML-based channel estimation [Gao+18; Sol+19; BDA20]. There also exist works about mapping learning in communication problems: specifically, the location/pseudo-location-to-beamformer mapping learning has been studied in [LeMag+22b] and [LeMag+22a], as well as the pseudo-location-to-best-codebook-index mapping learning in [Yas+23]. Additionally, our previous work in [Cha+24b] studied the location-to-channel mapping learning in a simplified scenario: only a single antenna and frequency were considered. This chapter builds on that foundation, offering a significant extension of the previously proposed method by incorporating support for both multi-antenna and multi-frequency scenarios, thereby addressing broader and more complex use cases. In [Zha+23], the authors propose to adapt the neural radiance fields concept of [Mil+21] from optical to radio-frequency signals. However, the proposed model requires complex learning strategies and long training times. Additionally, the proposed model is used for downlink channel estimation in a frequency division duplex setting, but requires the uplink channel knowledge. In this

chapter, the proposed approach only requires the receiver location knowledge and possesses relatively low sample complexity. Finally, in [Hoy+24], the authors proposed to calibrate electromagnetic properties of a scene, e.g. material permittivity and conductivity, scattering and reflection coefficients, using a differentiable ray-tracing approach. Upon completion of the calibration process, accessing the scene properties in different configurations becomes feasible. Among other applications, using the calibrated scene enables the accurate computation of the propagation channel response at different locations, through ray-tracing. The primary distinction between this differentiable ray-tracing method and the suggested approach is that the former emphasizes on calibrating a specific scene at the electromagnetic property level while considering the scene topology known. Conversely, the latter proposes a straightforward approach for the continuous location-to-channel mapping learning using a model-based neural network with no prior information about the scene topology.

Contributions

It is proposed to leverage the capabilities of model-based machine learning for the location-to-channel mapping learning. To maintain generality, this mapping learning problem is examined over multiple antennas and multiple frequencies. This chapter introduces the following contributions:

- A theoretical study of a physics-based channel model, using Taylor expansions, allows to clearly separate low and high-frequency spatial contents (Lemma 4.1, Proposition 4.1, Proposition 4.2).
- A sparse signal processing approach, using sparse representations of the channel, is presented to ensure the validity of the obtained channel approximation across the entire $\mathbb{R}^3$ space (Theorem 4.2).
- A model-based neural architecture is derived from the approximated channel model. It is shown that the obtained structure shares common features with classical INR architectures, such as a spectral separation stage with a high-frequency embedding, allowing to bypass the spectral-bias issue.
- Experiments are conducted on realistic channels, showing the great potential of the proposed approach, both in terms of learning performance and computational efficiency. Moreover, some experiments clearly show that the proposed approach successfully learns the underlying physical propagation model, enabling efficient generalization.

4.1.3 Problem formulation

Let us consider a communication system where a BS transmits information through N_a antennas and N_s distinct frequencies to mono-antenna UEs. In the time domain, the

propagation channel defines the filter operating on the electromagnetic waves transmitted between an emitting and receiving antenna [4]. This filter impulse response is classically known as the channel impulse response. Considering the propagation channel over L_p specular propagation paths between the jth BS antenna and a mono-antenna receiver located at $\mathbf{x} \in \mathbb{R}^3$ yields the following CIR definition:

$$h_j(t, \mathbf{x}) = \sum_{l=1}^{L_p} \alpha_{l,j}(\mathbf{x}) \, \delta(t - \tau_{l,j}(\mathbf{x})), \tag{4.4}$$

where $\alpha_{l,j}(\mathbf{x}) \in \mathbb{R}$, resp. $\tau_{l,j}(\mathbf{x}) \in \mathbb{R}$, is the attenuation coefficient, resp. propagation delay, for the lth path and jth BS antenna. Applying the Fourier transform on the CIR yields the channel frequency response. For a given frequency f_k, this gives:

$$h_j(f_k, \mathbf{x}) = \sum_{l=1}^{L_p} \alpha_{l,j}(\mathbf{x}) \, \mathrm{e}^{-\mathrm{j}2\pi f_k \tau_{l,j}(\mathbf{x})} \tag{4.5}$$

$$= \sum_{l=1}^{L_p} \frac{\gamma_l}{d_{l,j}(\mathbf{x})} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_k} d_{l,j}(\mathbf{x})}, \tag{4.6}$$

where $\lambda_k = c/f_k \in \mathbb{R}$ is the wavelength associated to frequency f_k and $d_{l,j}(\mathbf{x}) \in \mathbb{R}$ represents the propagation distance for the lth path and jth BS antenna. In further developments, the frequency dependence in Eq. (4.6) is dropped and the notation $h_{j,k}(\mathbf{x})$ is used. In Eq. (4.6), it is assumed that the attenuation is proportional to $d_{l,j}(\mathbf{x})$ and that the delay is such that:

$$\tau_{l,j}(\mathbf{x}) = \begin{cases} \frac{d_{l,j}(\mathbf{x})}{c} & \forall j, l = 1 \\ \frac{d_{l,j}(\mathbf{x})}{c} - \nu_{l,j} & \forall j, l \neq 1 \end{cases}, \tag{4.7}$$

where $\nu_{l,j} \in \mathbb{R}$ accounts for a supplementary delay induced by wave-matter interactions.

Remark 4.1. *Note that the attenuation model introduced in Eq.* (4.6)*, as well as the propagation delay model introduced in Eq.* (4.7)*, are a direct manifestation of the domain knowledge introduced in (H1).*

As a result of Eq. (4.7), the attenuation coefficient γ_l becomes complex: $\forall l > 1$, $\mathfrak{Re}\{\gamma_l\}$ and $\mathfrak{Im}\{\gamma_l\}$ represent the small-scale attenuation and phase shift introduced along the lth path. Note that $\mathfrak{Re}\{\gamma_1\} = 1$ and $\mathfrak{Im}\{\gamma_1\} = 0$ when considering a LoS path, as this path does not present any wave-matter interaction.

The contribution of reflected paths can be modeled using the virtual source theory [Poz98, Chapter 1, pp.47-49]. This theory states that a path, between a source and a receiver, reflected on a plane $\mathcal{R}$ can be equivalently represented as a direct path originating from the source's symmetric counterpart with respect to $\mathcal{R}$, and ending at the receiver, as illustrated in Fig. 4.2 (a). Moreover, while the virtual source theory is classically used to model reflections, it can also accommodate the modeling of diffraction [Liu+14]. In

4. See Chapter 1 for more information.

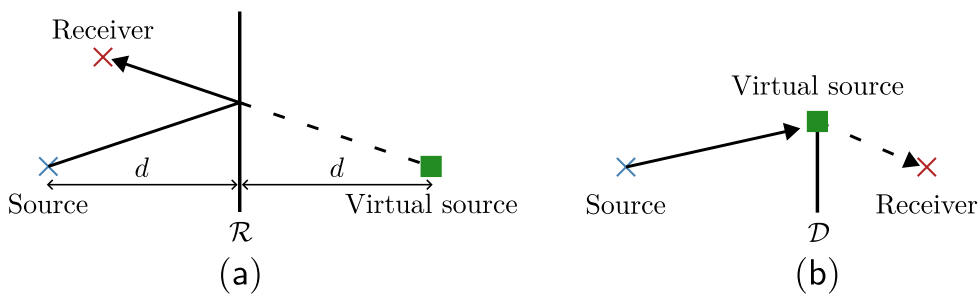


Figure 4.2: Virtual source theory used to model: (a) reflection on reflection plane $\mathcal{R}$, (b) diffraction on diffracting object $\mathcal{D}$.

this scenario, the virtual source is located at the endpoint of the diffraction object $\mathcal{D}$, as illustrated in Fig. 4.2 (b). Using this theory to model the propagation distance $d_l(\mathbf{x})$ yields:

$$h_{j,k}(\mathbf{x}) = \sum_{l=1}^{L_p} \frac{\gamma_l}{\|\mathbf{x} - \mathbf{a}_{l,j}\|_2} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_k}\|\mathbf{x}-\mathbf{a}_{l,j}\|_2}, \tag{4.8}$$

where $\mathbf{a}_{l,j} \in \mathbb{R}^3$ denotes the location of the lth virtual antenna, so that $1/\|\mathbf{x} - \mathbf{a}_{l,j}\|_2$ represents the large scale fading of the lth path.

Remark 4.2. *Note that Eq.* (4.8) *models a multipath channel, where the L_p propagation paths are distinguishable paths: they can result from reflection, diffraction or scattering.*

Remark 4.3. *The proposed channel model builds on the widely used ray-based channel model. This general framework subsumes specific models such as the Saleh-Valenzuela model, where path gains and ray directions follow specific statistical distributions. This general model provides greater flexibility to represent diverse channel scenarios, including indoor and outdoor environments, while also accounting for crucial physical propagation phenomena such as reflection and diffraction. This flexibility strengthens the general applicability of the proposed method.*

The exponential argument in Eq. (4.8) reveals a high-frequency spatial dependence: a small change in the location $\mathbf{x}$ induces a significant change in the channel coefficient $h_{j,k}(\mathbf{x})$. This arises from the wavelength dependence in the exponential argument: as the carrier frequency rises, the wavelength drops, yielding fast variations of channel coefficients in the location space. In classical communication systems (sub-6 GHz), λ is on the order of a few centimeters. 5G/6G systems also consider millimeter wavelengths.

Let $\mathbf{H}(\mathbf{x}) \in \mathbb{C}^{N_a \times N_s}$ be the antenna-frequency channel matrix at location $\mathbf{x}$. This matrix is constructed from the concatenation of the complex frequency channel coefficients $h_{j,k}(\mathbf{x})$ in Eq. (4.8), across every antenna and frequency. The goal of this study is to learn:

$$\begin{aligned} f_{\boldsymbol{\theta}} \colon \mathbb{R}^3 &\longrightarrow \mathbb{C}^{N_a \times N_s} \\ \mathbf{x} &\longrightarrow \mathbf{H}(\mathbf{x}), \end{aligned} \tag{4.9}$$

i.e. a neural network f parameterized by a set of learnable parameters $\boldsymbol{\theta}$ that learns the continuous mapping between a location $\mathbf{x}$ and complex channel matrix $\mathbf{H}(\mathbf{x})$. As previously mentioned, the high-frequency spatial dependency in the channel model causes the location-to-channel mapping to be remarkably hard to learn using classical neural architectures, due to the spectral-bias issue. Alternative neural architectures overcoming this issue are presented in the next section.

4.2 Towards a model-based approach

This section presents the proposed model-based approach, as well as the related theoretical results. Such theoretical developments are crucial for understanding the proposed model-based neural architecture presented in Section 4.3, as this architecture is based on Theorem 4.2. Although the use of Taylor approximations on the propagation distance is not novel, the contribution presented in this section is, to the best of the authors' knowledge, the first to propose an approximated channel model that enables extrapolation around both a reference receiver location and a reference emitter antenna location, relying solely on assumptions of attenuation and phase proportionality with the propagation distance. Additionally, this section presents a discussion on the use of sparse recovery methods to address the problem formulated in Eq. (4.20).

4.2.1 Model-based approach

It is proposed to use knowledge from the physics-based propagation model presented in Eq. (4.8), to derive a model-based neural architecture that can overcome the spectral-bias issue. As the high-frequency spatial dependency originates from the wavelength presence in the ratio $\|\mathbf{x}-\mathbf{a}_{l,j}\|_2/\lambda_k$ in Eq. (4.8), it is proposed to develop this term using a Taylor expansion. It will be shown that it allows to separate the high-frequency from the low-frequency spatial content. This approach, known as the plane wave approximation, is used to obtain the well-known steering vector model [TV05, Chapter 7], [LMLCP19].

Lemma 4.1: Local distance approximation

Let $\mathbf{x}_r \in \mathbb{R}^3$ be a reference location and $\mathcal{V}_{\mathbf{x}} \subset \mathbb{R}^3$ be a local validity domain such that $\forall \mathbf{x} \in \mathcal{V}_{\mathbf{x}}, \|\mathbf{x}-\mathbf{x}_r\|_2 \leq \epsilon_{\mathbf{x}}$. Let $\mathbf{a}_{l,r} \in \mathbb{R}^3$ be a reference antenna location and $\mathcal{V}_{\mathbf{a}} \subset \mathbb{R}^3$ be local validity domain such that $\forall \mathbf{a}_{l,j} \in \mathcal{V}_{\mathbf{a}}, \|\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\|_2 \leq \epsilon_{\mathbf{a}}$. One has, $\forall (\mathbf{x}, \mathbf{a}_{l,j}) \in \mathcal{V}_{\mathbf{x}} \times \mathcal{V}_{\mathbf{a}}$:

$$\|\mathbf{x}-\mathbf{a}_{l,j}\|_2 \simeq \|\mathbf{x}_r-\mathbf{a}_{l,r}\|_2 + \mathbf{u}_{l,j}(\mathbf{x}_r)^{\mathsf{T}}(\mathbf{x}-\mathbf{x}_r) - \mathbf{u}_{l,r}(\mathbf{x}_r)^{\mathsf{T}}(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}), \quad (4.10)$$

with $\mathbf{u}_{l,j}(\mathbf{x}_r) = (\mathbf{x}_r-\mathbf{a}_{l,j}) / \|\mathbf{x}_r-\mathbf{a}_{l,j}\|_2$.

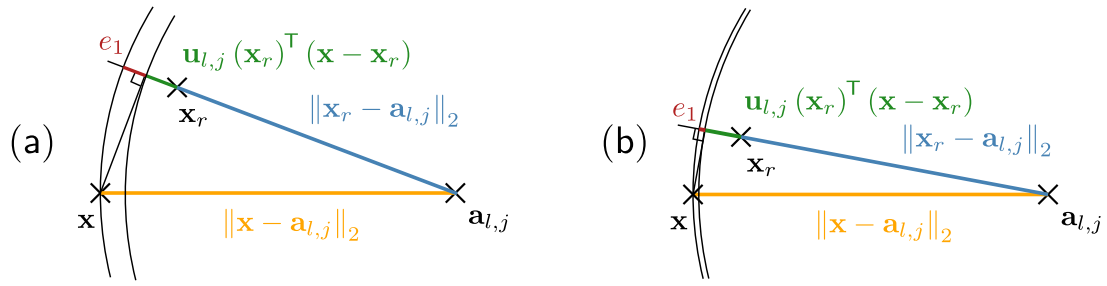


Figure 4.3: Taylor expansion on locations only: (a) location $\mathbf{x}$ far from reference $\mathbf{x}_r$, (b) location $\mathbf{x}$ close to reference $\mathbf{x}_r$. In both settings, e_1 characterizes the approximation error.

Proof. See Appendix D.1. ■

Fig. 4.3 illustrates the Taylor expansion on locations. One can remark the direct link between the approximation error and the distance to reference: when the considered location $\mathbf{x}$ is far from the reference $\mathbf{x}_r$, the approximation error is high, while on the other hand, when the distance to the reference is small, the approximation error decreases. This emphasizes the local validity of Lemma 4.1. A similar analysis can be made for the Taylor expansion on antennas.

Corollary 4.1 proposes to analytically characterize the approximation error of the Taylor expansion introduced in Lemma 4.1.

Corollary 4.1: Distance approximation error

Let $\mathbf{x}_r \in \mathbb{R}^3$ and $\mathbf{a}_{l,r} \in \mathbb{R}^3$ respectively denote a reference location and a reference antenna location. The approximation error in Eq. (4.10) can be approximated using the second order of the Taylor expansions as:

$$e \simeq \frac{1}{2}\left(\frac{\|\mathbf{x}-\mathbf{x}_r\|_2^2}{\|\mathbf{x}_r-\mathbf{a}_{l,j}\|_2} - o\left(\frac{1}{\|\mathbf{x}_r-\mathbf{a}_{l,j}\|_2^2}\right) + \frac{\|\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\|_2^2}{\|\mathbf{x}_r-\mathbf{a}_{l,r}\|_2} - o\left(\frac{1}{\|\mathbf{x}_r-\mathbf{a}_{l,r}\|_2^2}\right)\right). \tag{4.11}$$

Proof. See Appendix D.2. ■

One can remark in Corollary 4.1 that the approximation error is mostly dependent on the ratios $\|\mathbf{x}-\mathbf{x}_r\|_2^2 / \|\mathbf{x}_r-\mathbf{a}_{l,j}\|_2$ and $\|\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\|_2^2 / \|\mathbf{x}_r-\mathbf{a}_{l,r}\|_2$. In other words, when the distance between the reference location and current/reference antenna is important, the approximation error decreases. Indeed, in such scenario, the common far field approximation is valid, making the planar approximation of spherical wavefronts realistic. In Eq. (4.10), the planar wavefront term is represented by the first projection term: this term is null when the location $(\mathbf{x}-\mathbf{x}_r)$ is orthogonal to the direction $\mathbf{u}_{l,j}(\mathbf{x}_r)$. Once introduced in a complex exponential, it yields periodic parallel level sets in the $\mathbf{u}_{l,j}(\mathbf{x}_r)$

direction. Such phenomenon constitutes the definition of planar wavefronts. On the other hand, when the reference distances are small, the far field approximation does not hold, making the approximation error being directly dependent on the distances to the references $\|\mathbf{x}-\mathbf{x}_r\|_2$ and $\|\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\|_2$.

Proposition 4.1 presents the injection of the approximated propagation distance in the channel model in Eq. (4.8). This allows to obtain an approximated channel coefficient around a reference location and a reference antenna.

Proposition 4.1: Approximated channel interpretation

Let $\mathbf{x}_r \in \mathbb{R}^3$ and $\mathbf{a}_{l,r} \in \mathbb{R}^3$ be a reference location and reference antenna location. Let $d_{l,r} = \|\mathbf{x}_r - \mathbf{a}_{l,r}\|_2$, $\tau_{l,r} = d_{l,r}/c$, and $h_{l,r}(\mathbf{x}_r) = \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}d_{l,r}}/d_{l,r}$. Let $f_r \in \mathbb{R}$ be a reference frequency such that $\forall f_k \in \mathbb{R}, f_k = (f_k - f_r) + f_r$. One has, $\forall (\mathbf{x}, \mathbf{a}_{l,j}) \in \mathcal{V}_\mathbf{x} \times \mathcal{V}_\mathbf{a}$:

$$h_{j,k}(\mathbf{x}) \simeq \sum_{l=1}^{L_p} \underbrace{\gamma_l h_{l,r}(\mathbf{x}_r)}_{\text{Reference channel}} \underbrace{\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^\mathsf{T}(\mathbf{x}-\mathbf{x}_r)}}_{\text{Location correction}} \underbrace{\mathrm{e}^{-\mathrm{j}2\pi(f_k-f_r)\tau_{l,r}}}_{\text{Frequency correction}} \underbrace{\mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^\mathsf{T}\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)}}_{\text{Antenna correction}}. \tag{4.12}$$

Proof. See Appendix D.3. ■

It can be seen in Eq. (4.12) that $h_{j,k}(\mathbf{x})$ is a sum of reference channels, around a reference location, at a reference frequency, multiplied by location, frequency, and antenna correction terms. Letting $w_{l,r}(\mathbf{x}) = \gamma_l h_{l,r}(\mathbf{x}_r)\,\mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^\mathsf{T}\mathbf{x}_r} \in \mathbb{C}$, Eq. (4.12) can be further rearranged as, $\forall (\mathbf{x}, \mathbf{a}_{l,j}) \in \mathcal{V}_\mathbf{x} \times \mathcal{V}_\mathbf{a}$:

$$h_{j,k}(\mathbf{x}) \simeq \sum_{l=1}^{L_p} \underbrace{w_{l,r}(\mathbf{x})\,\mathrm{e}^{-\mathrm{j}2\pi(f_k-f_r)\tau_{l,r}}\mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^\mathsf{T}\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)}}_{\text{Slowly varying}} \underbrace{\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^\mathsf{T}\mathbf{x}}}_{\text{Fastly varying}}. \tag{4.13}$$

Remark 4.4. *One can observe in Eq.* (4.13) *that the channel coefficient* $h_{j,k}(\mathbf{x})$ *can be viewed as a linear combination of planar wavefronts. The directions of said planar wavefronts are defined by the spatial frequencies* $\mathbf{u}_{l,r}(\mathbf{x}_r)$*. Furthermore, Eq.* (4.13) *presents a spectral separation stage: the planar wavefronts present high-frequency spatial content because of their wavelength dependency in the exponential argument, while the weights, multiplied by the location and antenna-correction terms, present low-frequency spatial content (on the order of the Taylor-expansions validity domain).*

Proposition 4.2 presents the expansion of the obtained approximation to every antenna and frequency, i.e. expresses the channel matrix $\mathbf{H}(\mathbf{x})$ as a function of the location $\mathbf{x}$.

Proposition 4.2: Channel matrix local approximation

Let $\boldsymbol{\psi}_{\mathbf{a},l}(\mathbf{x}) \in \mathbb{C}^{N_a}$ be a steering vector (SV) and $\boldsymbol{\psi}_{\mathbf{f},l}(\mathbf{x}) \in \mathbb{C}^{N_s}$ be a frequency response vector (FRV). Let $\psi_{\mathbf{x},l}(\mathbf{x}) = \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^{\mathsf{T}}\mathbf{x}} \in \mathbb{C}$ be a planar wavefront. The multi-antenna multi-frequency channel can be expressed as, $\forall(\mathbf{x}, \mathbf{a}_{l,j}) \in \mathcal{V}_{\mathbf{x}} \times \mathcal{V}_{\mathbf{a}}$:

$$\mathbf{H}(\mathbf{x}) \simeq \sum_{l=1}^{L_p} w_{l,r}(\mathbf{x})\, \psi_{\mathbf{x},l}(\mathbf{x})\, \boldsymbol{\psi}_{\mathbf{a},l}(\mathbf{x}_r)\, \boldsymbol{\psi}_{\mathbf{f},l}(\mathbf{x}_r)^{\mathsf{T}}, \tag{4.14}$$

where

$$\boldsymbol{\psi}_{\mathbf{a},l}(\mathbf{x}_r) = \begin{bmatrix} \mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^{\mathsf{T}}\left(\mathbf{a}_{l,1}-\mathbf{a}_{l,r}\right)} \\ \vdots \\ \mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^{\mathsf{T}}\left(\mathbf{a}_{l,N_a}-\mathbf{a}_{l,r}\right)} \end{bmatrix} \in \mathbb{C}^{N_a}, \tag{4.15}$$

and

$$\boldsymbol{\psi}_{\mathbf{f},l}(\mathbf{x}_r) = \begin{bmatrix} \mathrm{e}^{-\mathrm{j}2\pi(f_1-f_r)\tau_{l,r}} \\ \vdots \\ \mathrm{e}^{-\mathrm{j}2\pi(f_{N_s}-f_r)\tau_{l,r}} \end{bmatrix} \in \mathbb{C}^{N_s}. \tag{4.16}$$

Proof. Direct derivation from Eq. (4.13) by defining SVs and FRVs. ■

The obtained results can be summarized as follows:

- Eq. (4.10) presents the Taylor expansion of the propagation distance around a reference location and reference antenna location.
- Eq. (4.13) presents how the Taylor expansions introduce a spectral separation stage in the propagation channel.
- Eq. (4.14) presents the expansion of the previous results over every antenna and frequencies.

One can remark that the approximation proposed in Eq. (4.14) is only valid in neighborhoods of the reference antenna location $\mathbf{a}_{l,r}$ and reference location $\mathbf{x}_r$, namely $\mathcal{V}_{\mathbf{a}}$ and $\mathcal{V}_{\mathbf{x}}$. Theorem 4.2 presents the expansion of the approximation validity domain to the entire $\mathbb{R}^3$ space. The idea is to partition $\mathbb{R}^3$ into location and antenna local validity domains, and then aggregate the needed planar wavefronts, SVs, and FRVs into dictionaries for each location and antenna local validity domain pair. Furthermore, the proof of Theorem 4.2 introduces the Direction of Departure (DoD) $\tilde{\mathbf{u}}_i$ so that the SV dictionary is constructed with only the physical antenna locations $\mathbf{a}_{1,j}$.

Theorem 4.2: Global sparse channel approximation

Let us consider the tiling of the location subset $\mathcal{S}_{\mathbf{x}} \subset \mathbb{R}^3$ into $\Omega_{\mathbf{x}}$ local validity domains. A second tiling is applied to the antenna subset $\mathcal{S}_{\mathbf{a}} \subset \mathbb{R}^3$ with $\Omega_{\mathbf{a}}$ local validity domains. Let $D \in \mathbb{N}^*$ such that $D \leq L_p \Omega_{\mathbf{x}} \Omega_{\mathbf{a}}$. Let $\tilde{\mathbf{\Psi}}_{\mathbf{a}} = \left[\tilde{\boldsymbol{\psi}}_{\mathbf{a},i}, \cdots, \tilde{\boldsymbol{\psi}}_{\mathbf{a},D}\right] \in \mathbb{C}^{N_a \times D}$ be a dictionary of SVs, defined as:

$$\tilde{\mathbf{\Psi}}_{\mathbf{a}} = \begin{bmatrix} \mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\tilde{\mathbf{u}}_1^{\mathsf{T}}(\mathbf{a}_{1,1}-\mathbf{a}_{1,r})} & \dots & \mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\tilde{\mathbf{u}}_D^{\mathsf{T}}(\mathbf{a}_{1,1}-\mathbf{a}_{1,r})} \\ \vdots & & \vdots \\ \mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\tilde{\mathbf{u}}_1^{\mathsf{T}}\left(\mathbf{a}_{1,N_a}-\mathbf{a}_{1,r}\right)} & \dots & \mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\tilde{\mathbf{u}}_D^{\mathsf{T}}\left(\mathbf{a}_{1,N_a}-\mathbf{a}_{1,r}\right)} \end{bmatrix}, \tag{4.17}$$

where $\tilde{\mathbf{u}}_i \in \mathbb{R}^3$ is a DoD. Let $\tilde{\mathbf{\Psi}}_{\mathbf{f}} = \left[\tilde{\boldsymbol{\psi}}_{\mathbf{f},1}, \cdots, \tilde{\boldsymbol{\psi}}_{\mathbf{f},D}\right] \in \mathbb{C}^{N_s \times D}$ be a dictionary of FRVs, defined as:

$$\tilde{\mathbf{\Psi}}_{\mathbf{f}} = \begin{bmatrix} \mathrm{e}^{-\mathrm{j}2\pi(f_1-f_r)\tau_1} & \dots & \mathrm{e}^{-\mathrm{j}2\pi(f_1-f_r)\tau_D} \\ \vdots & & \vdots \\ \mathrm{e}^{-\mathrm{j}2\pi(f_{N_s}-f_r)\tau_i} & \dots & \mathrm{e}^{-\mathrm{j}2\pi(f_{N_s}-f_r)\tau_D} \end{bmatrix}, \tag{4.18}$$

where $\tau_i \in \mathbb{R}^+$ is a propagation delay. Let $\tilde{\boldsymbol{\psi}}_{\mathbf{x}}(\mathbf{x}) = \left[\tilde{\psi}_{\mathbf{x},1}, \cdots, \tilde{\psi}_{\mathbf{x},D}\right] \in \mathbb{C}^D$ be a dictionary of planar wavefronts, defined as:

$$\tilde{\boldsymbol{\psi}}_{\mathbf{x}}(\mathbf{x}) = \left[\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_1^{\mathsf{T}}\mathbf{x}} \dots \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_D^{\mathsf{T}}\mathbf{x}}\right], \tag{4.19}$$

where $\mathbf{u}_i \in \mathbb{R}^3$ is a spatial frequency. Finally, let $\mathbf{w}(\mathbf{x}) \in \mathbb{C}^D$ be an activation vector such that: $\boldsymbol{\varpi}(\mathbf{x}) = \mathbf{w}(\mathbf{x}) \odot \tilde{\boldsymbol{\psi}}_{\mathbf{x}}(\mathbf{x})$. Then, $\forall \mathbf{x} \in \mathbb{R}^3$:

$$\mathbf{H}(\mathbf{x}) \simeq \sum_{i=1}^{D} \varpi_i(\mathbf{x})\, \tilde{\boldsymbol{\psi}}_{\mathbf{a},i} \tilde{\boldsymbol{\psi}}_{\mathbf{f},i}^{\mathsf{T}}. \tag{4.20}$$
$$\text{with } \|\boldsymbol{\varpi}(\mathbf{x})\|_0 = L_p$$

Proof. See Appendix D.4. ■

One can see that Theorem 4.2 proposes a sparse continuous interpolation of the channel matrix. The composite dictionary consists of all possible combinations of the required SVs and FRVs, while the activation vector presents spectral separation: it is defined with a low-frequency term $\mathbf{w}(\mathbf{x})$, and a high-frequency term $\tilde{\boldsymbol{\psi}}_{\mathbf{x}}(\mathbf{x})$.

Remark 4.5. *While Theorem 4.2 extends the validity expansion of Eq.* (4.14) *to the entire* $\mathbb{R}^3$ *space, the number of atoms* D *in each dictionary is intractable due to its dependence on the number of local validity domains* $\Omega_{\mathbf{x}}$ *and* $\Omega_{\mathbf{a}}$*. A way to overcome this issue is to*

construct each dictionary by discretizing its generating subspace, i.e. the DoD subspace, unit sphere $\mathcal{S}(\mathbf{0}_{\mathbb{R}^3}, 1)$*, for the SV dictionary, the delay space* $\mathbb{R}^+$ *for the FRV dictionary and the spatial frequency subspace,* $\mathcal{S}(\mathbf{0}_{\mathbb{R}^3}, 1)$*, for the planar wavefront dictionary. This approach is at the center of the model-based neural network architecture presented in Section 4.3.*

4.2.2 Discussion: a sparse recovery approach

As Eq. (4.20) can be viewed as a sparse reconstruction problem, one could think of finding the correct activation coefficients using sparse recovery techniques. Indeed, using matching pursuit/orthogonal matching pursuit (OMP) algorithms [MZ93; PRK93], one could find a sparse representation of $\mathbf{H}(\mathbf{x})$ in a composite dictionary of FRVs/SVs [5]. Obtaining a channel estimate $\hat{\mathbf{H}}(\mathbf{x})$ at a wanted location $\mathbf{x}$ could be separated into two stages. First, $\forall \mathbf{x}_r \in \mathcal{H}_r \subset \mathbb{R}^3$, $\mathcal{H}_r$ being a set of reference locations, one would have to solve:

$$\begin{aligned} \underset{\mathbf{w} = [w_1, \cdots, w_D]}{\text{minimize}} \quad & \left\| \mathbf{H}(\mathbf{x}_r) - \sum_{i=1}^{D} w_i \tilde{\psi}_{\mathbf{x},i}(\mathbf{x}_r) \tilde{\boldsymbol{\psi}}_{\mathbf{a},i} \tilde{\boldsymbol{\psi}}_{\mathbf{f},i}^{\mathsf{T}} \right\|_{\mathsf{F}}^2 . \\ \text{subject to} \quad & \mathbf{w} \in \mathbb{C}^D, \\ & \|\mathbf{w}\|_0 = L_p \end{aligned} \tag{4.21}$$

Then, for the wanted location $\mathbf{x}$, one could obtain the channel estimate as:

$$\hat{\mathbf{H}}(\mathbf{x}) = \sum_{i=1}^{D} w_{\mathbf{x}_{r,i}^\star} \tilde{\psi}_{\mathbf{x},i}(\mathbf{x}) \tilde{\boldsymbol{\psi}}_{\mathbf{a},i} \tilde{\boldsymbol{\psi}}_{\mathbf{f},i}^{\mathsf{T}}, \tag{4.22}$$

where $\mathbf{w}_{\mathbf{x}_r^\star} \in \mathbb{C}^D$ is the activation coefficient vector obtained by solving Eq. (4.21) for $\mathbf{x}_r^\star = \arg\min_{\mathbf{x}_r \in \mathcal{H}_r} \|\mathbf{x} - \mathbf{x}_r\|_2$, i.e. the closest reference location to the wanted location. However this approach presents an intractable complexity induced by the high-frequency nature of the propagation channel and the composite dictionary size. Indeed, as the propagation channel is a function rapidly varying with the location, one would have to consider spatially close reference locations, i.e. with spacing between reference locations under the wavelength. This introduces a high cardinality in $\mathcal{H}_r$, which increases the time complexity for achieving the first step of this estimation scheme. Additionally, as the composite dictionary is composed of every pair of FRVs/SVs, it results in a prohibitive dictionary size, further increasing the time complexity of the first estimation step.

The next section proposes to transform the obtained channel approximation in Eq. (4.20) into a neural architecture, following the model-based machine learning paradigm.

5. Note that a more detailed introduction to the matching pursuit technique is proposed in Chapter 6.

4.3 Model-based neural architecture

As it has been shown that classical signal processing methods are unsuited for solving the sparse reconstruction problem described in Theorem 4.2, it is proposed to leverage the learning capabilities of neural networks for the location-to-channel mapping learning. This section presents a model-based neural architecture derived from Eq. (4.20).

The approximation in Eq. (4.20) can be rewritten using the vectorization operator, clearly displaying the sparse representation structure of the approximated channel. Indeed, $\forall \mathbf{x} \in \mathbb{R}^3$:

$$\begin{aligned} &\operatorname{vec}\left(\mathbf{H}\left(\mathbf{x}\right)\right) \simeq \left(\tilde{\boldsymbol{\Psi}}_{\mathbf{f}}\left(\mathbf{x}\right) \otimes \tilde{\boldsymbol{\Psi}}_{\mathbf{a}}\left(\mathbf{x}\right)\right) \operatorname{vec}\left(\operatorname{diag}\left(\boldsymbol{\varpi}\left(\mathbf{x}\right)\right)\right), \\ &\text{with } \left\|\boldsymbol{\varpi}\left(\mathbf{x}\right)\right\|_0 = L_p \end{aligned} \tag{4.23}$$

where $\boldsymbol{\varpi}\left(\mathbf{x}\right) = \mathbf{w}\left(\mathbf{x}\right) \odot \tilde{\boldsymbol{\psi}}_{\mathbf{x}}\left(\mathbf{x}\right)$. As shown in Eq. (4.23), the location-to-channel mapping learning problem can be partitioned into four subproblems: the planar wavefronts, FRV, SV dictionaries, and the complex weights coefficients learning.

4.3.1 Main building blocks

Planar wavefront dictionary

As exposed by the previous analysis, the planar wavefronts take the form $\tilde{\boldsymbol{\psi}}_{\mathbf{x}}\left(\mathbf{x}\right) = \left[\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_1^\intercal\mathbf{x}}, \cdots, \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_D^\intercal\mathbf{x}}\right] \in \mathbb{C}^D$. A dictionary of planar wavefronts can then be constructed from the spatial frequencies $\mathbf{u}_i \in \mathbb{R}^3$, which by definition, are unit vectors representing the planar wavefronts direction. As a result, every spatial frequency belongs to a two-dimensional manifold: the unit sphere $\mathcal{S}\left(\mathbf{0}_{\mathbb{R}^3}, 1\right)$. Sampling $\mathcal{S}\left(\mathbf{0}_{\mathbb{R}^3}, 1\right)$ with D points yields the spatial frequency dictionary $\mathbf{u} = \left[\mathbf{u}_1, \cdots, \mathbf{u}_D\right] \in \mathbb{R}^{3\times D}$ that is used to compute the planar wavefront dictionary. The dictionary $\tilde{\boldsymbol{\psi}}_{\mathbf{x}}\left(\mathbf{x}\right)$ is then constructed using the Fourier feature (FF) layer, defined as:

$$\mathrm{FF} : \mathbf{x} \to \left[\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_1^\intercal\mathbf{x}} \quad \cdots \quad \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_D^\intercal\mathbf{x}}\right]. \tag{4.24}$$

Note that this FF layer is just a complex implementation of the cos/sin embedding layer used in RFF networks [RR07; Tan+20]. Also note that spatial frequencies could be learned either by gradient descent or directly through a neural network, however, they are kept fixed for this study.

FRV dictionary

It has been presented in Eq. (4.18) that the FRV dictionary $\tilde{\boldsymbol{\Psi}}_{\mathbf{f}}\left(\mathbf{x}\right) \in \mathbb{C}^{N_s\times D}$ only depends on the system frequencies f_k, $k \in [\![1, N_s]\!]$, the reference frequency f_r, and propagation delays τ_i. The system and reference frequencies are assumed to be known: such assumption is typically made in classical communication systems. In this study, the reference frequency

is computed as $f_r = \frac{1}{N_s}\sum_{k=1}^{N_s} f_k$. The FRV dictionary can then be constructed by sampling the propagation delay space, i.e. $\mathbb{R}^+$. In order to optimize the performance of the proposed approach, it is suggested to learn the discretization for every location. As such, it is proposed to use a MLP that learns a propagation delay vector $\boldsymbol{\tau}(\mathbf{x}) \in \mathbb{R}^D$ for every location $\mathbf{x}$. Note that, as the propagation delay vector exhibits slow variations in the location space, the MLP will not suffer from the spectral bias.

SV dictionary

As presented in Eq. (4.17), the SV dictionary $\tilde{\boldsymbol{\Psi}}_{\mathbf{a}}(\mathbf{x}) \in \mathbb{C}^{N_a \times D}$ only depends on the reference frequency wavelength λ_r, on the true antenna locations $\mathbf{a}_{1,j}$, on the true reference antenna location $\mathbf{a}_{1,r}$, and on DoDs $\tilde{\mathbf{u}}_i$. The reference frequency wavelength is assumed to be known, as well as the antenna locations. The true reference antenna location is computed as the barycenter of the true antenna locations. In order to construct the SV dictionary, one could discretize the DoD subspace $\mathcal{S}(\mathbf{0}_{\mathbb{R}^3}, 1)$. Similarly than for the FRV dictionary, it is proposed to optimize performance by learning the discretization for every location. As such, it is proposed to learn $\tilde{\mathbf{u}}(\mathbf{x})$ using a MLP, a method referred to as MB-$\tilde{\mathbf{u}}$ learning thereafter. An alternative approach consists in directly learning the entire SV dictionary for each location using a MLP, without assuming any dictionary structure. This approach, referred to as MB-$\tilde{\boldsymbol{\Psi}}_{\mathbf{a}}$ learning, releases constraints on the antenna correction terms at the expense of an increased learning-parameter complexity.

Complex weights learning

The complex activation weights $\mathbf{w}(\mathbf{x}) \in \mathbb{C}^D$ are learned following the same approach as in [Cha+24b]. More specifically, a MLP is used to learn the complex weight vector, $\mathbf{w}(\mathbf{x}) \in \mathbb{C}^D$, introducing the sparsity constraint with a `softmax` ponderation. Note that this hypernetwork encompasses the learning of the path attenuation. While the channel model presented in Eq. (4.6) assumes the attenuation being proportional to the propagation distance $d_l(\mathbf{x})$, there are cases where attenuation is proportional to $d_l(\mathbf{x})^\alpha$, with $\alpha \in \mathbb{R}^{+,*}$. The proposed neural model can handle such cases, as the MLP used for the complex activation weights learning is capable to learn such non-linear content. Finally, this MLP can adapt the sparsity of the learned activation vector, thereby effectively accounting for potential variations in the number of propagation paths for different locations within the considered scene.

4.3.2 Global architecture

The proposed model-based neural network architecture for the location-to-channel mapping learning is presented in Fig. 4.4. Note that this architecture only presents the MB-$\tilde{\mathbf{u}}$ approach, and that, during experiments, the used approach will be explicitly specified.

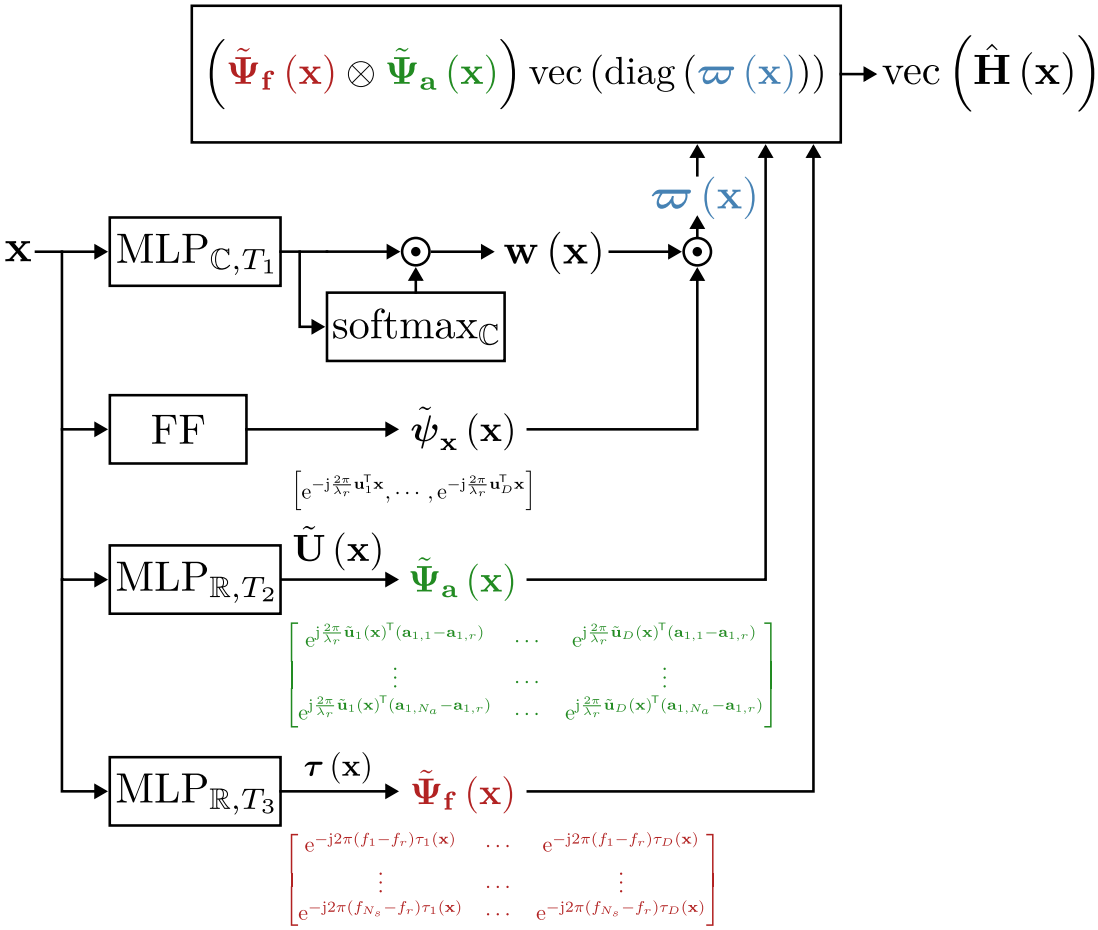


Figure 4.4: Proposed model-based neural architecture for the location-to-channel mapping learning.

Also note that $\mathrm{MLP}_{\mathbb{C},T}$ represents a 3-layer $\mathtt{ReLU}_{\mathbb{C}}$-MLP [6] with complex weights and biases, where each layer width is T. $\mathrm{MLP}_{\mathbb{R},T}$ represents the same MLP but with real weights/biases and $\mathtt{ReLU}$ [7] activation functions. The following remark emphasizes that the use of hypernetworks is a key feature of the proposed model.

Remark 4.6. *The proposed architecture shares common features with classical INR networks. Indeed, the FF layer can be seen as a positional-embedding stage, where the low dimensional location information is projected on a higher dimensional space containing high spatial frequencies. Nevertheless, the proposed architecture stands out from classical INR networks with the introduction of hypernetworks [Sch92; HDL17], i.e. parallel neural networks learning parameters of the main network. As illustrated in Fig. 4.4, four hypernetworks are used to learn parameters of the main branch: one for the complex activation vector learning* $\mathbf{w}(\mathbf{x})$*, and three dedicated for the planar wavefront* $\tilde{\boldsymbol{\psi}}_{\mathbf{x}}(\mathbf{x})$*, FRV* $\tilde{\boldsymbol{\Psi}}_{\mathbf{f}}(\mathbf{x})$ *and SV* $\tilde{\boldsymbol{\Psi}}_{\mathbf{a}}(\mathbf{x})$ *dictionaries constitution. Note that the proposed architecture is a specific instance of the hypernetwork architecture family, which includes the widely popular self-attention mechanism [Vas+17].*

4.4 Experiments

In this section, the mapping-learning performance of the proposed approach are evaluated on realistic synthetic data. Additionally, its performance is compared against classical architecture from the INR literature.

4.4.1 Learning framework and baselines

Learning framework

The dataset consists of locations and associated channel coefficients. Its generation can be divided into two phases:

1. Location generation.
2. Channel computation at the generated locations.

System parameters. For all datasets, the central frequency, i.e. the reference frequency f_r, is set to 3.5 GHz and the considered bandwidth is 50 MHz. The BS is equipped with a uniform linear array (ULA) with half reference wavelength spacing. The number of considered antennas N_a and frequencies N_s is presented for each experiment.
Location generation. For all datasets, a L by L square scene, with $L = 10$ m, is considered as the location space. Inside this scene, training locations are uniformly sampled with a certain location density. The uniform sampling influences the location generation process:

6. $\forall z_1 \in \mathbb{C}, \forall \mathbf{z}_2 \in \mathbb{C}^N$, $\mathtt{ReLU}_{\mathbb{C}}(z_1) = \mathtt{ReLU}(\mathfrak{Re}\{z_1\}) + \mathtt{jReLU}(\mathfrak{Im}\{z_1\})$, and $\mathtt{softmax}_{\mathbb{C}}(\mathbf{z}_2) = \mathtt{softmax}(|\mathbf{z}_2|)$.
7. $\mathtt{ReLU}(x) = \max(0, x)$.

when considering N training locations, the training location matrix can be expressed as $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{N\times 2}$, with:

$$\forall i \in [\![1, N]\!],\ \mathbf{x}_i \sim \mathcal{U}\left([-L/2, L/2] \times [-L/2, L/2]\right). \tag{4.25}$$

The spatial density impacts the number of generated locations within the scene. The L by L square scene covers an area S_m m^2, or equivalently S_λ λ^2. To achieve a desired spatial density of γ_m locs./m^2, or equivalently γ_λ locs./λ^2, the following formula is applied:

$$N = S_m \gamma_m = S_\lambda \gamma_\lambda. \tag{4.26}$$

Test locations are generated across a 2D uniform grid within the scene. The spacing between each test location in both the x and y directions is set to $\lambda/4$. In the considered scene and at the considered reference frequency, this grid yields around 210k test locations, enabling a thorough assessment of the mapping learning capabilities. Additionally, the spatial proximity of test locations allows to assess the learning performance of small scale channel fading.

Remark 4.7. *Note that while all previous theoretical developments are proven in the $\mathbb{R}^3$ scenario, the following simulations for the location-to-channel mapping learning capabilities are performed in $\mathbb{R}^2$ due to computational complexity. Hence, for all datasets, the locations are considered on the same elevation plane as the BS, i.e. considering bidimensional locations and azimuth DoDs. Consequently, spatial frequencies are sampled from $\mathcal{C}\left(\mathbf{0}_{\mathbb{R}^2}, 1\right)$, i.e. from the unit circle.*

Channel generation. Realistic synthetic channels are generated using the Sionna ray-tracing library [Hoy+22]. For each training/test location, ray-tracing techniques find the propagation paths and compute the associated channel coefficients. Three different datasets are considered:

- $\mathtt{D}_1$: complex dataset with variable number of propagation paths regarding the considered location in the scene.
- $\mathtt{D}_2$: $\mathtt{D}_1$ with no LoS path.
- $\mathtt{D}_3$: LoS dataset.

$\mathtt{D}_2$ enables investigation of the mapping-learning capabilities of the proposed model under conditions characterized by an equitable distribution of path contributions, where no single path exhibits dominant power. As depicted in Fig. 4.5, $\mathtt{D}_1$ and $\mathtt{D}_2$ are generated in the *Etoile* Sionna scene in Paris, while $\mathtt{D}_3$ is generated in an empty scene. For all datasets, diffraction and scattering are not considered. In $\mathtt{D}_1$ and $\mathtt{D}_2$, each propagation path can present at most two consecutive reflections.

Training loss and evaluation metric. All networks are trained using the classical ℓ_F loss, defined as:

$$\mu = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, \mathbf{H}(\mathbf{x})) \in \mathcal{D}} \left\| \mathbf{H}\left(\mathbf{x}\right) - f_{\boldsymbol{\theta}}\left(\mathbf{x}\right) \right\|_{\mathsf{F}}^2, \tag{4.27}$$

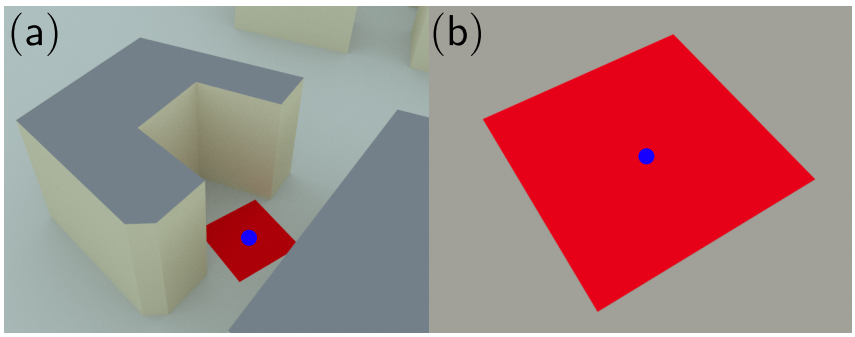


Figure 4.5: Ray-tracing scenes in Sionna: the red plane represents the possible training/test locations and the blue dot represents the BS. (a): scene used for $\mathtt{D}_1$ and $\mathtt{D}_2$. (b): scene used for $\mathtt{D}_3$.

where $f_{\boldsymbol{\theta}}(\mathbf{x}) = \hat{\mathbf{H}}(\mathbf{x})$ is the output of the neural network, and $\mathcal{D} \subset \mathbb{R}^2 \times \mathbb{C}^{N_a \times N_s}$ is the training location/channel dataset for the considered scene. The evaluation metric is the normalized mean squared error (NMSE) in dB over the test grid, defined as:

$$\mathsf{NMSE}_{(\mathrm{dB})} = 10 \log_{10} \left(\frac{1}{|\mathcal{T}|} \sum_{(\mathbf{x}, \mathbf{H}(\mathbf{x})) \in \mathcal{T}} \frac{\|\mathbf{H}(\mathbf{x}) - f_{\boldsymbol{\theta}}(\mathbf{x})\|_{\mathsf{F}}^2}{\|\mathbf{H}(\mathbf{x})\|_{\mathsf{F}}^2} \right), \tag{4.28}$$

where $\mathcal{T} \subset \mathbb{R}^2 \times \mathbb{C}^{N_a \times N_s}$ is the test location/channel dataset for the considered scene.

Remark 4.8. *While the proposed classical ℓ_F loss function may appear overly simplistic for this learning task, the subsequent experiments demonstrate that it achieves satisfactory mapping learning results. A more complex loss function, which would incorporate the selection of the correct planar wavefronts/SVs/FRVs for each training location, could be envisioned. However, in contrast to the proposed ℓ_F loss, such approach would require a more extensive data collection process, as both the azimuth and elevation angle of arrivals would need to be gathered for each training location.*

Highlight 4.2: Comparison with ray-tracing.

Given that the proposed model necessitates a training phase utilizing channel coefficients, obtained through ray-tracing in this study, a natural point of inquiry emerges regarding the rationale for not directly employing ray-tracing for channel prediction, given its applicability as an applied AI model.

The primary theoretical and practical advantage of the proposed method over ray-tracing is that accurate channel prediction through ray-tracing necessitates the constitution of a digital twin, with rigorous modelization of the scene at the electromagnetic level. Specifically, this ray-tracing approach mandates the determination of each scene element's electromagnetic properties such as material

permittivity or conductivity and reflection/diffraction coefficients. In contrast, the proposed method exclusively requires the collection of channel coefficients at different locations within a scene, and demonstrates independence from a priori knowledge of the scene topology. Furthermore, the ray-tracing approach exhibits a significantly superior computational complexity: predicting the channel at a given location involves shooting rays in all directions across the $\mathbb{R}^3$ space. Conversely, once trained, the proposed method exclusively involves computationally efficient matrix operations, as described in Eq. (4.23).

Baselines

It is proposed to assess the performance of the proposed architecture against baseline models from the INR literature. As such, several baselines are proposed in Fig. 4.6: (a) is a classical complex MLP while (b) and (c) are complex RFFs. Additionally, in (b), the spatial frequencies are drawn from a gaussian distribution, while (c) considers the same spatial frequencies as in the model-based architecture, i.e. sampled from the unit circle $\mathcal{C}\left(\mathbf{0}_{\mathbb{R}^2}, 1\right)$. All MLPs output a complex vector of dimension $N_a N_s$ which is then reshaped to obtain the estimated channel matrix $\hat{\mathbf{H}}\left(\mathbf{x}\right) \in \mathbb{C}^{N_a \times N_s}$.

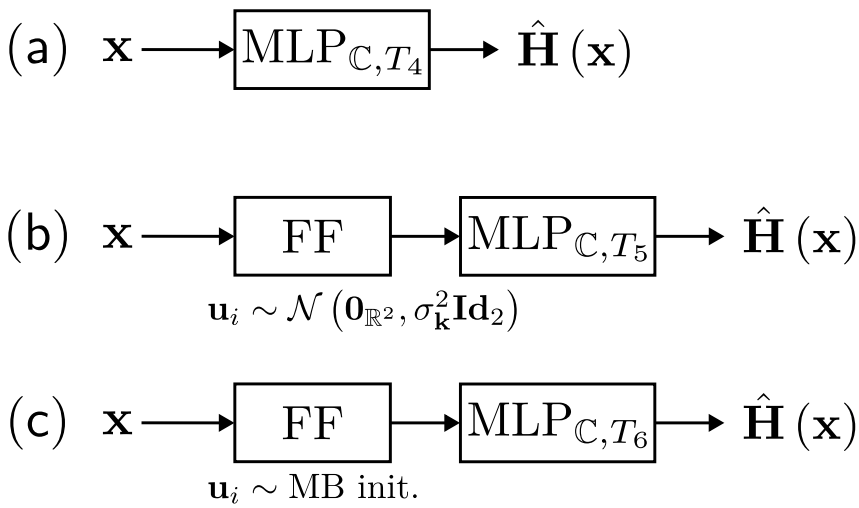


Figure 4.6: Baselines: (a) MLP, (b) and (c) RFFs.

4.4.2 Results and applications

Experimental results

For all experiments, the proposed networks and baselines have the following layer sizes: $T_1 = 256$, $T_2 = T_3 = 64$, $T_4 = 1024$, and $T_5 = T_6 = 64$. Additionally, unless stated

otherwise, $D = 1000$ spatial frequencies are considered. Those values have been empirically chosen to maximize performance and minimize overfitting. For RFFs, increasing the number of learnable parameters did not improve performance. Finally, a training location density of 175 locs./m^2 $\simeq$ 1.3 locs./λ^2, which is equivalent to 17.5k training locations, is considered at the exception of the experiment depicted in Fig. 4.8.

Scene reconstruction. Let $N_a = 64, N_s = 64$. It is proposed to study the ability of each network to learn the location-to-channel mapping in different radio-environments.

Table 4.1 presents the reconstruction results. One can remark that the proposed MB-$\tilde{\mathbf{\Psi}}_\mathbf{a}$ model outperforms every baseline and the MB-$\tilde{\mathbf{u}}$ model in every propagation scene. The performance gap observed between the MB-$\tilde{\mathbf{\Psi}}_\mathbf{a}$ and MB-$\tilde{\mathbf{u}}$ models is due to a hypothesis made in the proof of Proposition 4.1, and is explained in a subsequent experiment. The inferior mapping learning capabilities of the baselines can be explained by the lack of structure in their architecture.

Fig. 4.7 presents the reconstructed channel after training, for several models. The top row represents the real part of the reconstructed channels, while the bottom row represents the associated spatial spectrums, i.e. 2D Fourier transform. Note that the vertical and horizontal lines in the bottom row are artifacts arising from the rectangular windowing of the scene. It clearly appears that the MB network efficiently learns the desired mapping. Additionally, one can remark that the RFF model presents high-frequency spatial contents due to its embedding stage, as shown on the bottom row, but fails to learn the complex structure of the propagation channel. Finally, the reconstructed channel for the MLP model presents low-frequency spatial contents: its spectral content is concentrated near the origin on the bottom row. This behavior illustrates the spectral-bias issue inherent to this architecture.

Fig. 4.8 presents the mapping-learning performance evolution with respect to the training location density on the $\mathtt{D}_1$ dataset. One can remark that the MB-$\tilde{\mathbf{\Psi}}_\mathbf{a}$ network outperforms the baseline in any density configuration. Additionally, the proposed network presents a failure mode in the low location density regime. In this regime, the scarcity of spatially close training locations results in the learning failure of the rapidly varying spatial content. Note that the proposed method achieves quasi perfect reconstruction in sub-Shannon-Nyquist location density, as the 2D Shannon-Nyquist criterion yields a density of 4 locs./λ^2.

Fig. 4.9 presents the ground-truth and estimated frequency/antenna responses for a given test location in the $\mathtt{D}_1$ dataset. One can see that, due to the multipath nature of the $\mathtt{D}_1$ scene, the frequency response presents rapid variations. Both the estimated frequency and antenna responses are close to the ground-truth highlighting the very good reconstruction performance of the proposed model network.

Generalization capabilities. Fig. 4.10 presents the frequency generalization performance of the MB-$\tilde{\mathbf{\Psi}}_\mathbf{a}$ model trained on the $\mathtt{D}_1$ dataset with $N_s = 32$, and evaluated on $N_s = 64$ frequencies. The dashed line in Fig. 4.10 represents the training/test frequency separation. The NMSE, computed over the test locations, at the central antenna for each frequencies, is unsurprisingly higher for the frequencies unseen during training but still remains low.

Table 4.1: NMSE (in dB) over the test grid for the different datasets.

	MLP	RFF	RFF (MB init.)	**MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$**	MB-$\tilde{\mathbf{u}}$
Learnable parameters	10.5M	669k	669k	**9.1M**	851k
$\mathtt{D}_1$	0.01	0.02	2.10^{-3}	$\mathbf{-29.23}$	-14.60
$\mathtt{D}_2$	0.01	0.05	3.10^{-3}	$\mathbf{-20.19}$	-10.25
$\mathtt{D}_3$	0.01	0.02	2.10^{-3}	$\mathbf{-40.67}$	-11.40

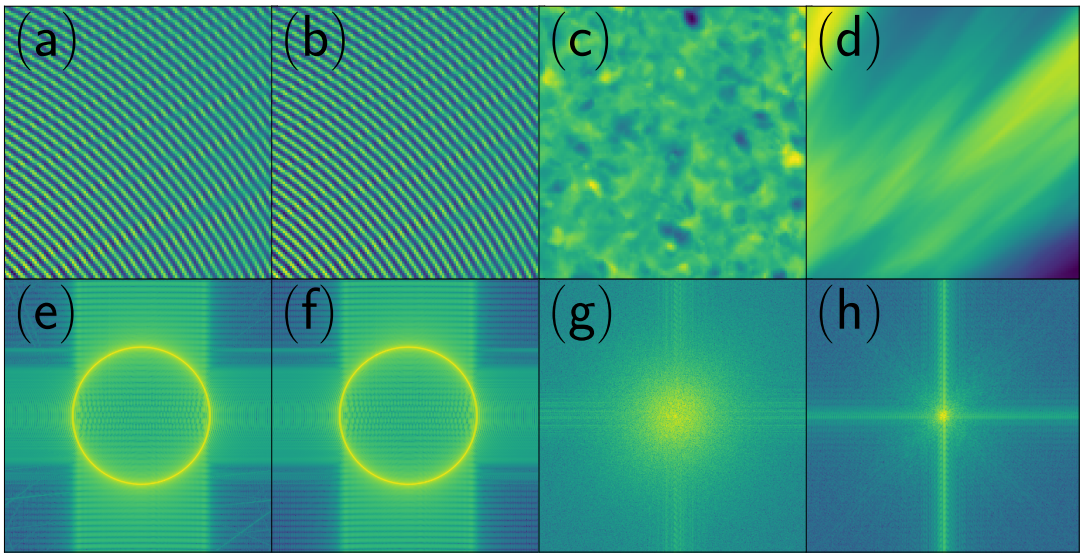


Figure 4.7: Top row: real part of reconstructed channel over a 2.5m by 2.5m zone of $\mathtt{D}_1$, for the central antenna and frequency ($N_a = 64$, $N_s = 64$). Bottom row: spatial spectrum of the reconstructed channel over the entire $\mathtt{D}_1$ scene (logarithmic scaling), for the central antenna and frequency ($N_a = 64$, $N_s = 64$). (a)/(e): Ground Truth, (b)/(f): MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$, (c)/(g): RFF, (d)/(h): MLP.

It highlights the generalization capabilities of the model-based neural network: as the FRV hypernetwork learns propagation delays, it successfully learns the physics of wave propagation in the considered scene, and is agnostic to the considered frequencies. This is illustrated on the right side of Fig. 4.10, where the reconstructed channel for a given location $\mathbf{x}_1$ is almost perfect on the frequencies used during training, and still follows the ground-truth for frequencies not seen during training.

Radio-environment compression. Let $N_a = 64, N_s = 64$. Let us consider the transmission, or storage, of the test dataset: in addition to the location information, one has to transmit $2N_aN_sN_l$ real numbers where N_l is the number of locations in the test dataset. As the test location grid is very dense ($N_l \simeq 210$k), it could be more efficient to only trans-

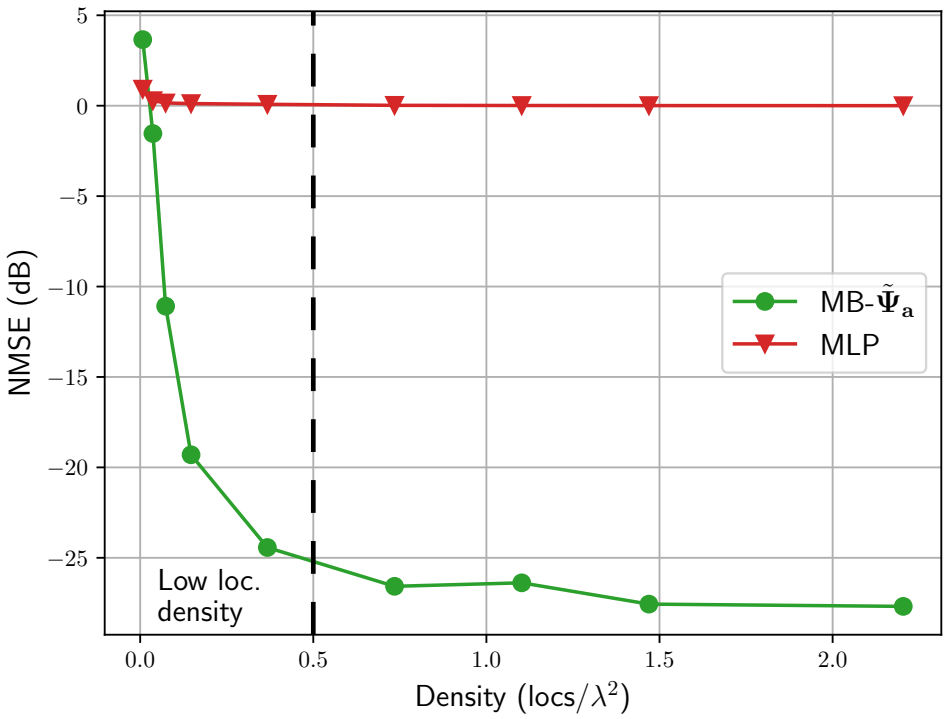


Figure 4.8: NMSE evolution with the training location density: MB-$\tilde{\mathbf{\Psi}}_\mathbf{a}$ learning, $\mathtt{D}_1$ dataset ($N_a = 64$, $N_s = 64$).

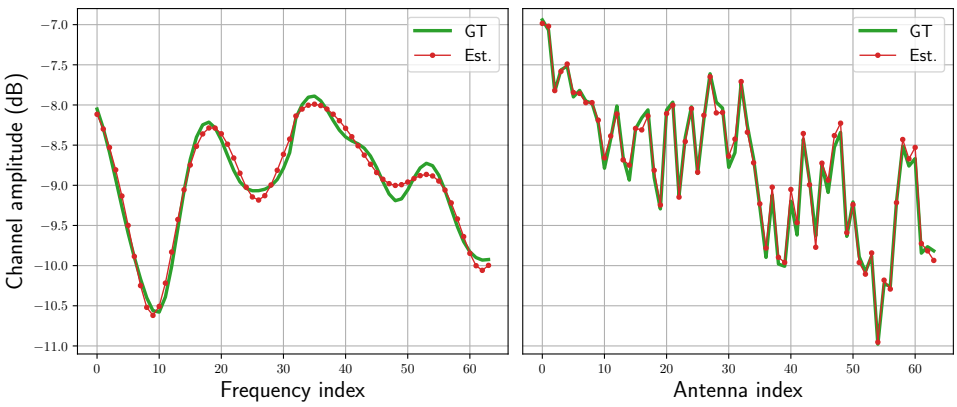


Figure 4.9: Reconstruction performance for a given location $\mathbf{x}_0$: MB-$\tilde{\mathbf{\Psi}}_\mathbf{a}$ learning, $\mathtt{D}_1$ dataset ($N_a = 64$, $N_s = 64$).

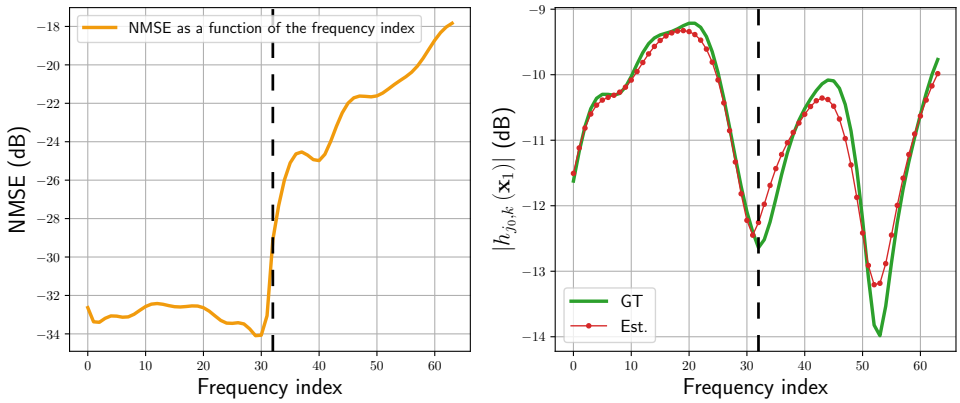


Figure 4.10: Frequency generalization performance: MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ learning, $\mathtt{D}_1$ dataset ($N_a = 64$, $N_s = 32$ for training, $N_s = 64$ for testing).

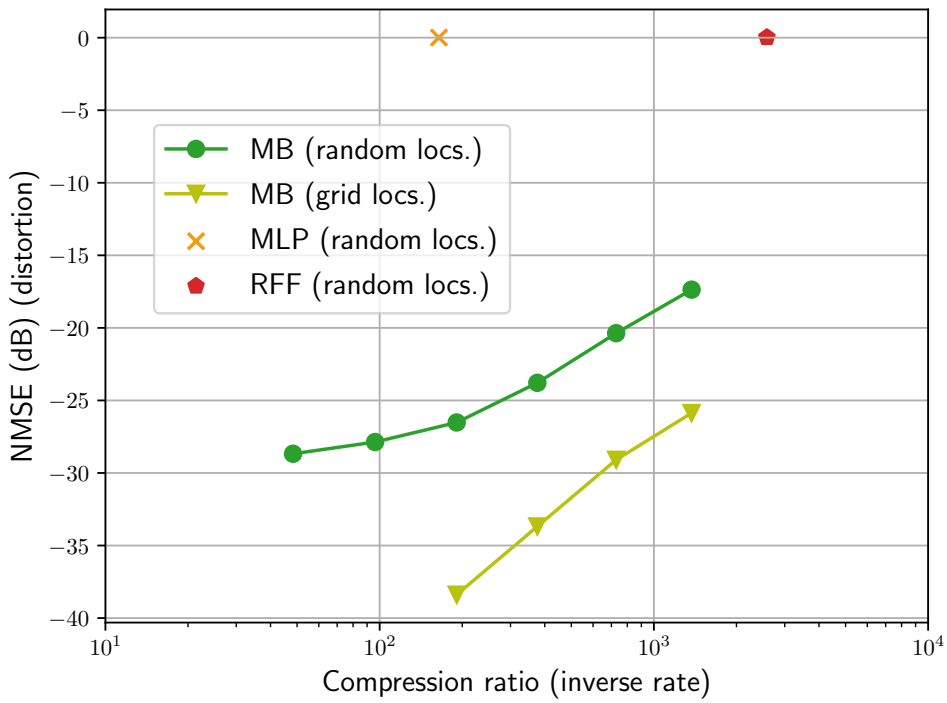


Figure 4.11: Learning performance evolution with the compression ratio: MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ learning, $\mathtt{D}_1$ dataset ($N_a = 64$, $N_s = 64$).

mit the weights of the trained model-based neural architecture. Indeed, as the MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ model achieves good location-to-channel mapping learning, one can use this network and the transmitted locations to reconstruct the channel. As an example, when considering the encoding of variables using 32 bits, the channel coefficients over the test grid weight around 6.9 Go while the 9.1M learnable parameters of the MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ model (with $D = 1000$) only weight around 36.4 Mo. Let N_b be the number of real learnable coefficients of the proposed neural architecture, the compression ratio is then defined as:

$$\gamma = \frac{2N_a N_s N_l}{N_b}. \tag{4.29}$$

Fig. 4.11 presents the evolution of the learning performance with respect to the compression ratio γ for various networks. It can be shown that the number of sampled spatial frequencies D is predominant on the learning error for the MB models. Therefore, it is proposed to only vary this parameter to vary N_b. The random locs. approach refers to the training on random locations with a spatial density of 1.3 locs./λ^2, while the grid locs. refers to the training on the entire $\lambda/4$ uniform grid. One can remark in Fig. 4.11 that, even at high compression ratios, the NMSE of the MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ model remains significantly low, highlighting the efficiency of the proposed approach for radio-environment compression. One can also note that, when considering the grid locs. approach, the model-based neural network reaches very good reconstruction performance, with a NMSE performance increase by around 10dB in comparison to the random locs. approach. Therefore, it enables the transmission of the dataset through the network weights with minimal reconstruction error, at the expense of a longer training phase. One should note that the compression ratio is virtually infinite. Indeed, as the trained network efficiently learns the location-to-channel mapping, it is able to infer the channel matrix for any location in the considered scene with good performance, yielding an infinite N_l in Eq. (4.29). Finally, one can interpret this experiment under the rate-distortion paradigm: in this context, the distortion measure is represented by the NMSE in dB, while the rate is represented as the inverse of the compression ratio. As the compression ratio increases, i.e. the rate decreases, the proposed neural model is required to encode the channels using fewer learnable parameters, resulting in a higher NMSE, which indicates an increase in distortion. This behavior aligns with the rate-distortion theory, where the trade-off between distortion and rate is investigated. In Fig. 4.11, it can be seen seen that the distortion is nearly logarithmically proportional to the rate inverse. This implies that a slight decrease in rate, or equivalently, an increase in the compression ratio, results only in a modest increase in distortion.

Performance evolution with N_a and N_s. Fig. 4.12 exposes the performance evolution along the antenna array and system frequencies for the MB-$\tilde{\mathbf{u}}$ learning network, trained on the $\mathtt{D}_1$ dataset with $N_a = 64$, $N_s = 64$. It presents the NMSE along the antenna indexes, computed over the test grid at the central frequency, and the NMSE along the frequency indexes, computed over the test grid at the central antenna. One can observe that, while the NMSE is almost constant over the different frequencies, it rises on the antenna array sides. This can be explained through the previous theoretical developments:

the model-based neural network has been designed with two Taylor expansions, one on the locations and another on the antennas. When the considered antenna array is large, the antenna correction terms fail for the array sides due the local nature of the Taylor expansion. This results in an increased error response on the antenna array sides. On the other hand, as the frequency correction terms do not originate from an approximation [8], the error response over the different system frequencies does not exhibit error spikes. It motivates the use of the MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ model which drops the model-based constraints for the FRV dictionary, overcoming this side effect. Note that a more extensive comparison between the two architectures is presented in a following experiment.

Fig. 4.13 presents the performance evolution of the MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ model with respect to the number of frequencies, by setting $N_a = 2$ and letting N_s vary over the fixed 50 MHz bandwidth. One can remark that the NMSE curve presents a minor performance degradation when N_s increases from 1 to 8 and then remains constant for higher values of N_s. As the FRV hypernetwork is independent of the number of frequencies, increasing N_s is equivalent to considering a harder learning problem with the same number of learnable parameters. This explains the minor performance degradation when N_s increases. However note that, even with $N_s = 64$, the NMSE performance remains satisfactory. Additionally, the adaptability of the proposed architecture is enhanced: in [Cha+24b], the proposed network was only able to learn the location-to-channel mapping for a unique antenna and frequency. Directly transposing the model-based architecture of [Cha+24b] for multi-antenna/multi-frequency would require one parallel model for each wanted antenna/frequency pair. The proposed model-based architecture directly scales with the number of frequencies: as the FRV hypernetwork is independent of N_s, one can use the same network for any wanted number of frequencies with the same learnable parameter complexity. The only complexity difference comes from Kronecker product of Eq. (4.23), which is negligible compared to the total numbers of operations in the neural network.

MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ learning vs MB-$\tilde{\mathbf{u}}$ learning. During the theoretical developments of Proposition 4.1, the hypothesis of DoD equality over the entire antenna array was made to obtain Eq. (4.12). This hypothesis fails in two scenarios, presented in Fig. 4.14: when the considered scene is close to the antenna array, and when the considered scene is far from a large antenna array.

Fig. 4.15 presents the performance evolution with respect to the number of antennas for $N_s = 2$ and both the MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ and MB-$\tilde{\mathbf{u}}$ models trained on the $\mathtt{D}_3$ dataset. One can remark that the NMSE increases with the number of antennas for the MB-$\tilde{\mathbf{u}}$ model: this is an immediate consequence of the aforementioned failure scenarios. Indeed, as the propagation scene for the $\mathtt{D}_3$ dataset consists of a small zone around the BS, when one increases the number of antennas, the assumption of DoD equality across all antennas does not hold anymore, resulting in a performance loss. This phenomenon is absent for the MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ model: in this architecture, the entire FRV matrix is learned, overcoming the DoD assumption issue.

8. See Eq. (D.22) in Appendix D.3.

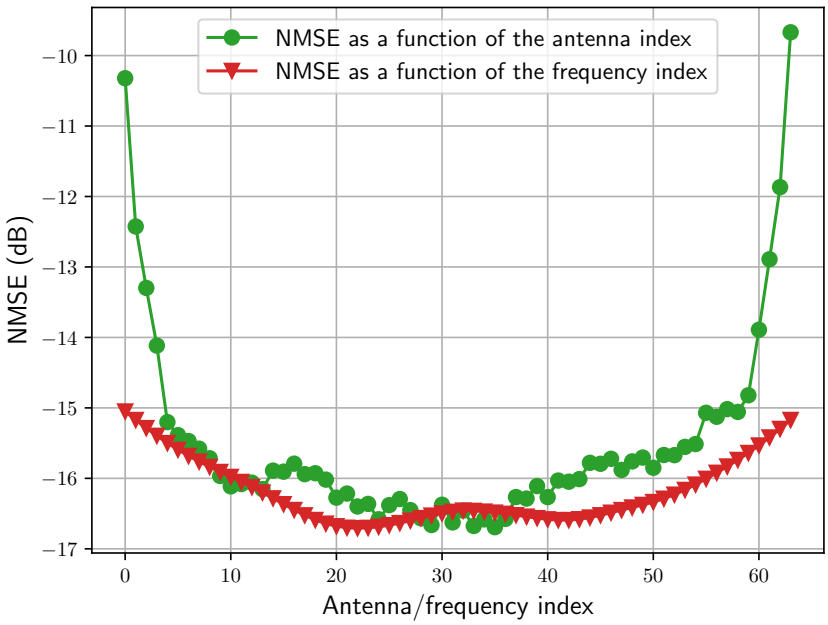


Figure 4.12: NMSE evolution with antenna/frequency index: MB-$\tilde{\mathbf{u}}$ learning, $\mathtt{D}_1$ dataset ($N_a = 64$, $N_s = 64$).

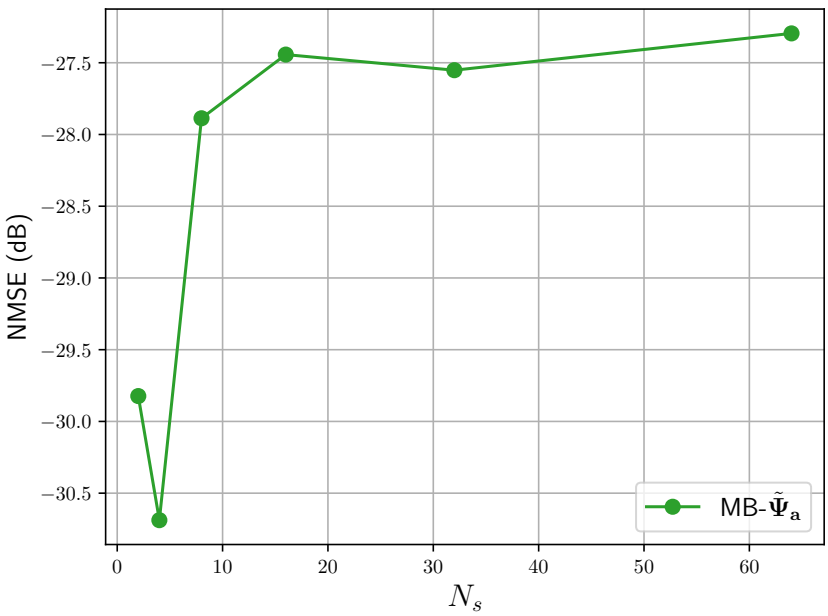


Figure 4.13: NMSE evolution with the number of frequencies N_s: MB-$\tilde{\mathbf{\Psi}}_{\mathbf{a}}$ learning, $\mathtt{D}_1$ dataset ($N_a = 2$).

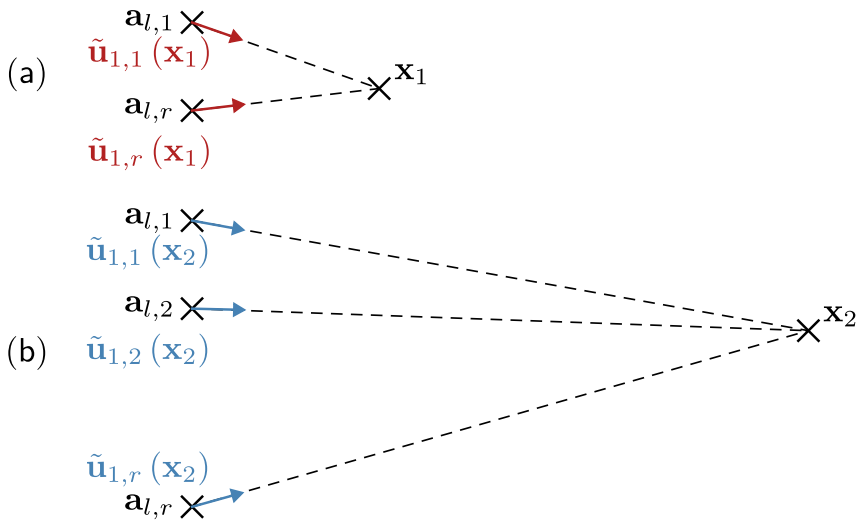


Figure 4.14: Failure scenarios for the DoD equality hypothesis: (a) locations close to the antenna array, (b): locations far from a large antenna array.

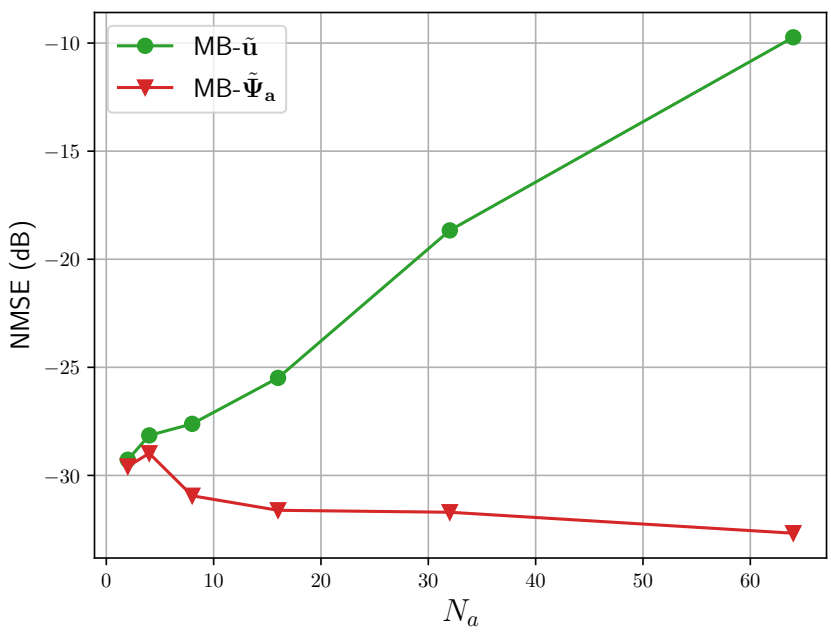


Figure 4.15: NMSE evolution with N_a: $\mathtt{D}_3$ dataset ($N_s = 2$).

Areas of applications

Previous experiments assumed perfect knowledge of the location where the channel needed to be evaluated. However, if a location offset arises from an imperfect location estimation process, the predicted channel from the trained model will correspond to the channel at the offset location. Since the channel varies rapidly with the location, on the order of the wavelength, the error between the true channel and the predicted channel will be significant, thereby limiting the application of the proposed method for precise channel estimation. Notwithstanding, the proposed method can be considered for different applications, some of which are outlined below.

Radio-environment compression. As discussed in Section 4.4.2, the model's ability to learn the location-to-channel mapping enables the storage of channel coefficients within its learnable parameters. Fig. 4.11 illustrates the reconstruction performance as a function of the compression ratio, demonstrating that the test dataset can be compressed up to a compression ratio of 10^3 without significant reconstruction error.

UE localization. Let $\mathbf{x} \in \mathbb{R}^3$ be a UE physical location and $\mathbf{H}(\mathbf{x}) \in \mathbb{C}^{N_a \times N_s}$ the corresponding channel coefficients. One might inquire the following question: *Is it possible to estimate* $\mathbf{x}$ *using solely the trained neural architecture and* $\mathbf{H}(\mathbf{x})$*?* This question is answered in Chapter 5, where it is shown that the proposed model-based neural architecture enables radio-localization with sub-wavelength accuracy.

General communication tasks. Upon training completion, the BS could use this neural model to obtain a rough estimate of the channel coefficients for different UEs, without the need of sending pilot symbols. This could be useful for many tasks: efficient beam management, resource allocation, or interference management.

4.5 Conclusion

This chapter focused on the application of the MB-ML paradigm to the location-to-channel mapping learning task, formalized in Problem 3.1. In this context, several key observations can be drawn:

- Domain knowledge, formalized in (H1), was leveraged to derive a propagation channel model. Analytical developments based on this model yielded an approximate location-to-channel map exhibiting a sparse dictionary structure. This approximated map was translated into a neural network architecture, where compact hypernetworks were trained to learn the individual dictionaries, preserving the overall sparse organization. Thus, domain knowledge was used to **structure** the learning method, specifically by guiding the design of a neural network architecture.

- Experiments conducted on realistic synthetic data, and comparisons with baselines from the INR literature, established the superior performance of the proposed MB-ML model, as well as its ability to overcome the spectral bias issue. Furthermore, its application to Problem 3.1, i.e. radio-environment compression, was illustrated.

Transmitting the parameters of the trained model-based neural architecture, and using it to reconstruct the channel coefficients at wanted locations has been shown to be significantly more efficient than directly transmitting the channel coefficients, achieving compression ratios of up to 10^3 with minimal performance degradation.

- The interpretability benefits of the MB-ML paradigm were also emphasized. Theoretical developments used to derive the approximated channel model provided insights on performance of the model-based architecture across various scenarios.

Chapter 5

Location-to-channel mapping: localization application

This chapter investigates the use of the neural architecture introduced in Chapter 4 as a neural generative model in a localization context. In particular, it examines how this architecture can enhance the performance of fingerprinting-based localization methods, achieving sub-wavelength accuracy in several realistic propagation environments. Section 5.1 provides a formal introduction to the radio-localization problem. Section 5.2 details the proposed model-based method, and introduces theoretical performance bounds. Section 5.3 presents an experimental validation of the proposed approach. Finally, Section 5.4 offers concluding remarks.

Contents

5.1 Introduction **144**

5.1.1 General introduction 144

5.1.2 Contributions and related work 145

5.1.3 System model and problem formulation 147

5.2 Proposed approach **148**

5.2.1 Learning a generative neural channel model 148

5.2.2 Application to localization 151

5.2.3 About the channel function injectivity 153

5.2.4 Performance and complexity optimization 156

5.3 Experiments **160**

5.3.1 Evaluation framework 160

5.3.2 Experimental results 162

5.4 Conclusion **170**

This chapter is mainly based on a journal article submitted to the *IEEE Transactions of Wireless Communications* [Cha+25b]. Furthermore, the experiments described in this chapter entailed a total computational cost of approximately 145 GPU days.

5.1 Introduction

5.1.1 General introduction

Radio localization refers to the process of determining the location of an object, such as a transmitter, a receiver or a passive target, based on measurements of radio frequency electromagnetic waves. One of the earliest application of radio localization is radio detection and ranging (radar), which has been extensively used in both military, e.g. detecting adversarial aircrafts, and civilian contexts, e.g. air traffic control in civil aviation. Traditional radio localization techniques rely on the use of signal processing methods such as angle of arrival, time of arrival, time difference of arrival or round trip time: see [Liu+07] for an extensive review. These methods typically require multiple sensing nodes and assume the existence of a direct LoS propagation path with negligible NLoS paths treated as nuisance. However, in complex radio environments, NLoS paths may be significant, resulting in degraded localization accuracy.

The evolution towards the sixth-generation of cellular networks has introduced the concept of integrated sensing and communication, wherein sensing tasks, such as localization, and communication functions are jointly developed to improve system efficiency and performance. This places the localization problematic at the core of future cellular communication systems, as these methods will be extensively used to enable a wide range of localization-based applications [DiTar+14; Wym+20]. From this perspective, it is crucial to develop localization methods that achieve high precision in complex radio propagation environments, in order to maximize the efficiency of future applications. One example of a radio-localization-aided application is the deployment of automated guided vehicles in smart factories, wherein precise positioning is essential for efficient and autonomous operations. Smart factories are typically complex indoor environments with challenging propagation conditions, i.e. quasi-static channel but with significant NLoS paths, where classical localization methods often suffer from degraded performance.

Over the past decade, rapid developments in the machine learning field have facilitated the development of learning-based localization methods, particularly those leveraging fingerprinting. These methods exploit patterns in a radio-environment related feature, e.g. received signal strength or channel coefficients, to perform localization by associating a given measured feature with a position, through the use of a ML method. Such approach has been studied by the 3GPP [3GP21; 3GP22]. In this context, it was reported that, in complex indoor radio environments, the 90% quantile localization error metric could be reduced from approximately 15 m with classical methods to within the range of $1 - 5$ m when leveraging ML-aided techniques [Cha+23]. However, these approaches still rely on channel estimates from multiple base stations, and the proposed neural architectures often introduce significant computational complexity during both training and inference phases, in addition to requiring large dataset sizes. This chapter investigates to what extent the model-based machine learning paradigm [Shl+23; SE23], identified as a key strategy for improving the efficiency and interpretability of ML techniques, can be leveraged for accurate and efficient radio localization. Given the well-known challenges of localization

in complex radio environments, several key questions arise: *Does a model-based machine learning localization method achieve superior performance in challenging radio environments compared to classical methods? Does it entail a lower computational complexity than other machine-learning methods?*

5.1.2 Contributions and related work

Contributions

Our previous studies [Cha+24b; Cha+25a] introduced a novel neural architecture [1] that combines the concept of implicit neural representation [2], with the model-based machine learning paradigm, in order to learn the location-to-channel mapping at the wavelength level. The neural network proposed in [Cha+25a], and presented in Chapter 4, is capable of inferring the channel coefficients between a multi-antenna base station operating over multiple frequencies and a mono-antenna user equipment at a given location, achieving high accuracy. Once trained, this neural network can be used as a generative neural channel model on a given scene. This chapter investigates the use of this neural model to perform localization with sub-wavelength precision. The main contributions can be summarized as follows.

- A localization method that leverages the learned location-to-channel mapping to perform data augmentation within a fingerprinting framework is introduced in Definition 5.3 and illustrated in Fig. 5.1.
- The performance and computational complexity of the proposed method are optimized in Section 5.2.4, using theoretical insights introduced in Corollary 5.2.
- Several experiments conducted on realistic channels generated through ray-tracing in both outdoor and complex indoor environments demonstrate the sub-wavelength precision and computational efficiency of the proposed method.

Related work

ML-based localization methods is a well studied topic in the recent literature: the reader is referred to [Nes+20] and [Bur+20] for comprehensive surveys. Fingerprinting-based localization can rely on a wide range of ML methods: support vector machine [Far+11], k-NN [Hoa+18], decision trees [Yim08], random forests [Qin+20], principal component analysis [Sal+16], but also deep neural networks [BRY20; Cha+23].

On the other hand, the use of the model-based machine learning paradigm has been applied to several wireless communication problems such as precoding [LS23], detection [SDW17], channel estimation [He+18; WHD21; YLM22; CLMR23], angle of arrival

1. See Chapter 4 for a detailed presentation.
2. See [Ess+25] for an extensive literature review on the use of INRs.

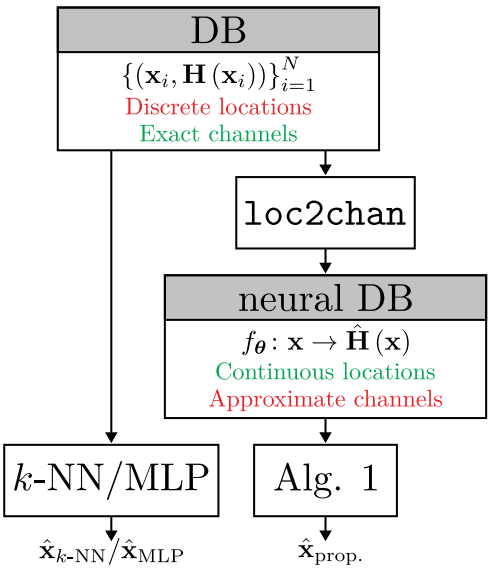


Figure 5.1: Overview of the proposed method: the left-hand side illustrates the classical k-NN/MLP approaches using a fixed database (DB), whereas the right-hand side depicts the proposed scheme, in which a neural network learning the location-to-channel mapping serves as a neural database.

estimation [Shm+25; Cha+25c], channel charting [Yas+22; Yas+23; Yas+24; Cha+24a], but also in integrated sensing and communication scenarios [MatRam+24; MatRam+25b]. Its use in a localization context has been studied in [Yan+21], where a model-based neural network is employed to estimate the position and velocity of a moving UE. Specifically, these characteristics are obtained by solving a weighted least squares problem, where the weight matrix is dynamically adapted to the input measurements through a neural network. The primary distinction between [Yan+21] and the approach proposed in this chapter lies in methodology: while [Yan+21] leverages data to dynamically solve a conventional least-squares problem, the proposed method exploits prior knowledge of the channel propagation model to perform data augmentation and guide the optimization process. Furthermore, this study considers a static UE and performs localization using only uplink channel coefficients obtained at a single BS, whereas [Yan+21] addresses a mobile UE and relies on measurements from multiple BSs processed centrally. Finally, to the best of the author's knowledge, the work presented in this chapter is the first to explore the use of a generative neural channel model to perform localization. While a simplified evaluation setup with noiseless, static channels is considered, this work provides an initial validation of the proposed method, and still reflects relevant real-world scenarios. Indeed, the noiseless assumption is justified by the widespread use of pilot symbols in sensing systems. Additionally, while static or quasi-static propagation environments are not representative of all propagation environments, there exist contexts where this assumption is reasonable, e.g. in certain indoor environments, such as indoor factories where precise localization supports automation. In time-varying environments, periodic data collection

combined with re-training of the proposed neural architecture could further extend the presented approach.

5.1.3 System model and problem formulation

Let us consider a scene where the location space is a plane denoted by $\mathcal{S} \subset \mathbb{R}^2$. The considered task is the localization of a mono-antenna UE located at $\mathbf{x} \in \mathcal{S}$, which transmits pilots on N_s distinct frequencies to a BS equipped with N_a antennas, using only the noiseless uplink channel matrix $\mathbf{H}(\mathbf{x}) \in \mathbb{C}^{N_a \times N_s}$. It is assumed that the considered propagation scene is static.

Remark 5.1. *As mentioned in the introduction, the assumptions of noiseless and static channels are physically grounded and reflect practical scenarios, especially indoor environments such as factories, where precise localization is critical for automation.*

This localization problem can be formally defined as determining a location estimator $\mathsf{E}^\star : \mathcal{D}_f \to \mathcal{S}$, that minimizes the mean localization error, i.e.:

$$\mathsf{E}^\star = \arg\min_{\mathsf{E} \in \mathcal{X}} \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}} \left[\| \mathbf{x} - \mathsf{E}\left(f\left(\mathbf{H}(\mathbf{x})\right)\right) \|_2 \right], \tag{5.1}$$

where $\mathcal{X}$ is the location estimator class, $f(\mathbf{H}(\mathbf{x}))$ is a channel feature, $\mathcal{D}_f = \mathrm{dom}(f)$ being its domain. Additionally, $p_{\mathbf{x}}$ denotes the true locations distribution over $\mathcal{S}$. The challenge lies in developing an estimator capable of providing accurate location estimates across the entire scene, with minimal a-priori knowledge. One classical approach in the localization literature is fingerprinting with propagation channel coefficients, which can be summarized as follows:

1. In a given propagation scene, one constitutes a dictionary of known UE locations $\mathcal{G}$:

$$\mathcal{G} = \{\tilde{\mathbf{x}}_i\}_{i=1}^{N_f} \subset \mathcal{S}, \tag{5.2}$$

   and a dictionary of the corresponding propagation channel coefficients $\mathcal{H}$:

$$\mathcal{H} = \{\mathbf{H}(\tilde{\mathbf{x}}_i)\}_{i=1}^{N_f} \subset \mathbb{C}^{N_a \times N_s}, \tag{5.3}$$

   where $\mathbf{H}(\tilde{\mathbf{x}}_i) \in \mathbb{C}^{N_a \times N_s}$ is the uplink channel matrix between the BS and UE located at $\tilde{\mathbf{x}}_i \in \mathcal{S}$, and N_f is the number of collected location/channel coefficient pairs.

2. For the UE with unknown location $\mathbf{x}$, the BS estimates the uplink channel matrix $\mathbf{H}(\mathbf{x})$. The estimated location is subsequently obtained as:

$$\hat{\mathbf{x}}(\mathbf{H}(\mathbf{x})) = \arg\max_{\tilde{\mathbf{x}} \in \mathcal{G}} \mathrm{sim}\left(\mathbf{H}(\mathbf{x}), \mathbf{H}(\tilde{\mathbf{x}})\right), \tag{5.4}$$

with $\mathrm{sim}(\cdot,\cdot) : \mathbb{C}^{N_a \times N_s} \times \mathbb{C}^{N_a \times N_s} \to]-\infty, 1]$ being a channel similarity measure, optimal at 1 [3].

This approach exhibits two shortcomings. First, for the estimated location to be unique within $\mathcal{G}$, the channel function needs to be injective with respect to the similarity measure, as defined below.

Definition 5.1: Channel function injectivity

The channel function is said to be injective wrt. similarity measure iff:

$$\forall (\mathbf{x}, \tilde{\mathbf{x}}) \in \mathcal{S} \times \mathcal{S}, \mathrm{sim}(\mathbf{H}(\mathbf{x}), \mathbf{H}(\tilde{\mathbf{x}})) = 1 \Rightarrow \mathbf{x}_m = \mathbf{x}_n. \tag{5.5}$$

Second, the precision of the estimated location is directly related to the spatial density of the location subset $\mathcal{G}$. Indeed, assuming that the true locations are uniformly distributed, a manifestation of the well-known curse of dimensionality [HTF09, Chapter 2, pp.22-26] reads as:

$$\mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}} \left[\| \mathbf{x} - \hat{\mathbf{x}}(\mathbf{H}(\mathbf{x})) \|_2 \right] \propto \left(\frac{A_{\mathcal{S}}}{|\mathcal{G}|} \right)^{\frac{1}{d}}, \tag{5.6}$$

where $\hat{\mathbf{x}}(\mathbf{H}(\mathbf{x}))$ is obtained by solving Eq. (5.4), $A_{\mathcal{S}}$ is the area covered by $\mathcal{S}$, and $d = \dim_{\mathbb{R}}(\mathcal{S}) = 2$. Enhancing the localization accuracy thus necessitates an extensive measurement process, which poses challenges in both the time required for dictionary construction, and the memory requirements for storing the dictionary. For a fixed covered area $A_{\mathcal{S}}$, dividing the localization error by a factor $n \in \mathbb{R}^*$ requires multiplying the cardinality of $\mathcal{G}$ by a factor n^d. This results in an impractically large dataset size, which represents the main limitation of classical fingerprinting methods.

5.2 Proposed approach

This section presents how a model-based neural network learning the location-to-channel mapping can be used to overcome the limitations of the classical fingerprinting-based localization method.

5.2.1 Learning a generative neural channel model

It is proposed to use the neural architecture presented in [Cha+24b], and expanded in [Cha+25a] [4], as a generative neural channel model. Considering L_p propagation paths,

3. Note that the choice of the similarity function's codomain is abitrary.
4. Note that this neural architecture constitutes the main contribution of Chapter 4.

the channel coefficients at frequency f_k between antenna j of the BS and the UE located at $\mathbf{x} \in \mathcal{S}$ can be modeled as:

$$h_j\left(f_k, \mathbf{x}\right) = \sum_{l=1}^{L_p} \frac{\gamma_l}{d_{l,j}\left(\mathbf{x}\right)} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_k} d_{l,j}(\mathbf{x})}, \tag{5.7}$$

where $\gamma_l \in \mathbb{C}$ accounts for wave-matter interactions of NLoS paths, $d_{l,j}\left(\mathbf{x}\right)$ is the length of the lth propagation path for the jth BS antenna, and $\lambda_k = c/f_k$ [5].

Remark 5.2. *Using the virtual source theory [Poz98, Chapter 1, pp.47-49], [Liu+14], one can rewrite Eq. (5.7) as:*

$$h_j\left(f_k, \mathbf{x}\right) = \sum_{l=1}^{L_p} \frac{\gamma_l}{\left\|\mathbf{x} - \mathbf{a}_{l,j}\right\|_2} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_k} \left\|\mathbf{x} - \mathbf{a}_{l,j}\right\|_2}, \tag{5.8}$$

where $\mathbf{a}_{l,j} \in \mathbb{R}^2$ is the jth true antenna location, when $l = 1$, or jth virtual antenna location, when $l > 1$. This channel model will be used in the subsequent theoretical developments.

Let $\mathbf{H}\left(\mathbf{x}\right) \in \mathbb{C}^{N_a \times N_s}$ be the uplink channel matrix at location $\mathbf{x}$, i.e. the concatenation of the channel coefficients defined in Eq. (5.8), for all antennas and frequencies. In [Cha+25a], the authors designed a model-based neural network $f_{\boldsymbol{\theta}}$ from the propagation model presented in Eq. (5.8), $\boldsymbol{\theta}$ being its set of learnable parameters. This neural network learns the mapping:

$$\begin{aligned} f_{\boldsymbol{\theta}} \colon \mathbb{R}^2 &\longrightarrow \mathbb{C}^{N_a \times N_s} \\ \mathbf{x} &\longrightarrow \mathbf{H}\left(\mathbf{x}\right), \end{aligned} \tag{5.9}$$

by minimizing the loss function:

$$\mu = \frac{1}{\left|\mathcal{D}\right|} \sum_{(\mathbf{x}, \mathbf{H}(\mathbf{x})) \in \mathcal{D}} \left\|\mathbf{H}\left(\mathbf{x}\right) - f_{\boldsymbol{\theta}}\left(\mathbf{x}\right)\right\|_{\mathsf{F}}^2, \tag{5.10}$$

over the training dataset $\mathcal{D} \subset \mathcal{S} \times \mathbb{C}^{N_a \times N_s}$, consisting of locations and associated uplink channel matrices.

Remark 5.3. *In [Cha+25a], all theoretical developments are made for true locations $\mathbf{x} \in \mathbb{R}^3$. This study only considers the $\mathbb{R}^2$ scenario to simplify the associated theoretical analysis. This assumption is equivalent to having all UEs confined to the same elevation plane as the BS. It is worth noting that all the proposed theoretical results can easily be extended to $\mathbb{R}^3$.*

In this study, $f_{\boldsymbol{\theta}}$ is the MB-$\tilde{\boldsymbol{\Psi}}_{\mathbf{a}}$ neural network proposed in [Cha+25a], whose architecture is recalled in Fig. 5.2. For a given input location $\mathbf{x}$, the associated channel $\mathbf{H}\left(\mathbf{x}\right)$ is infered

5. Eq. (5.7) accounts only for direct propagation, reflection, and diffraction. Note that scattering effects can be incorporated by decoupling the distances used in the attenuation term (as a product of distances) and in the phase term (as a sum of distances).

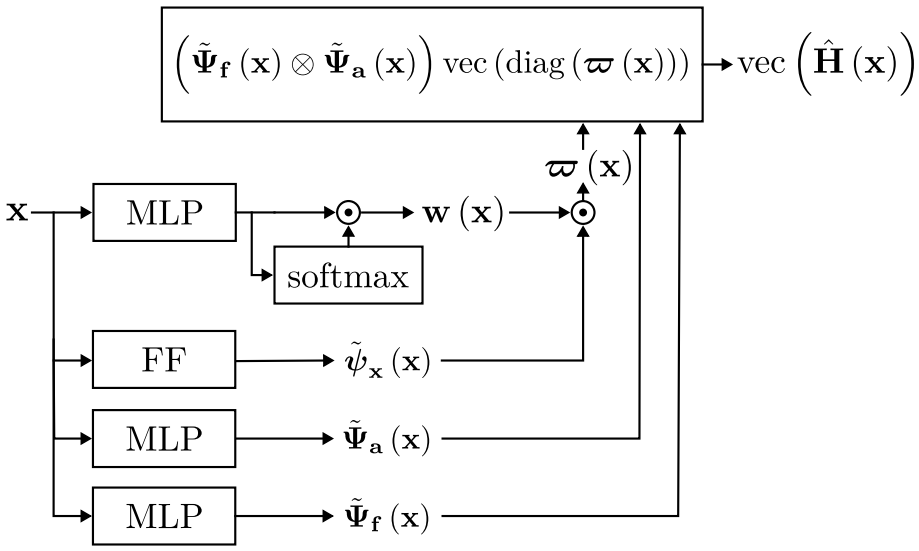


Figure 5.2: Architecture of the neural model $f_{\boldsymbol{\theta}}$ proposed in [Cha+25a]. The Fourier feature (FF) embedding projects the input location $\mathbf{x}$ into a subspace that captures high frequency variations. This figure can be seen as a simplified version of Fig. 4.4.

as a sparse linear combination of steering vectors (denoted by the dictionary $\tilde{\mathbf{\Psi}}_{\mathbf{a}}(\mathbf{x})$), of frequency response vectors (denoted by the dictionary $\tilde{\mathbf{\Psi}}_{\mathbf{f}}(\mathbf{x})$) and planar wavefronts (denoted by the dictionary $\tilde{\boldsymbol{\psi}}_{\mathbf{x}}(\mathbf{x})$), which accounts for local displacements. Each MLP consists of a distinct three-layer MLP hypernetwork, with hidden layer sizes of 256 for the weight hypernetwork and 64 for the antenna and frequency hypernetworks. In total, the model comprises about 9.1 million learnable parameters. It was shown that, after training, this neural architecture was able to infer the channel matrix at any wanted location in the scene considered during training, with minimal error, yielding the following definition.

Definition 5.2: Model estimation error

Let $\mathcal{S} \subset \mathbb{R}^2$ be the set of locations covered by the considered scene, and $f_{\boldsymbol{\theta}}$ the trained neural model. Then, the model estimation error $\Xi(\mathbf{x})$ is bounded as, $\exists \epsilon_{f_{\boldsymbol{\theta}}} \in \mathbb{R}^+$ s.t. $\forall \mathbf{x} \in \mathcal{S}$:

$$\Xi(\mathbf{x}) = \frac{\|\mathbf{H}(\mathbf{x}) - f_{\boldsymbol{\theta}}(\mathbf{x})\|_{\mathsf{F}}}{\|\mathbf{H}(\mathbf{x})\|_{\mathsf{F}}} \leq \epsilon_{f_{\boldsymbol{\theta}}}. \tag{5.11}$$

The estimation error $\Xi(\mathbf{x})$ of the proposed network remains low, even in complex radio environments. In [Cha+25a], the NMSE on the test dataset, defined as $\mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}}\left[\Xi(\mathbf{x})^2\right]$, was reported to be approximately -40 dB in LoS-only environments, -29 dB in LoS/NLoS

environments, and -20 dB in NLoS-only environments [6]. This good performance on the test dataset indicates that the proposed network learns the underlying propagation physics rather than memorizing the training data. Furthermore, the proposed network was shown to effectively capture the fast-fading characteristics of the channel, i.e. small-scale fading, allowing it to infer distinct channel matrices for locations separated by sub-wavelength distances, while maintaining low estimation error. These findings support the feasibility of using $f_{\boldsymbol{\theta}}$ as a generative neural channel model within the considered scene.

Remark 5.4. *Once the location-to-channel mapping is learned through $f_{\boldsymbol{\theta}}$, it can be used to generate any number of approximated channels at any given location in the scene, allowing to generate a potentially infinite database to be used for localization. Moreover, the mapping being continuous and differentiable, first order optimization method can be applied. These two properties are at the core of the proposed method, which is described in details next.*

5.2.2 Application to localization

As presented in Eq. (5.6), the primary drawbacks of fingerprinting-based localization methods include the need for a substantial comparison dictionary, which in turn necessitates an extensive measurement campaign and imposes memory storage constraints. It is proposed to use the proposed model-based neural architecture to solve this dictionary size issue. Indeed, as $f_{\boldsymbol{\theta}}$ can be seen as a generative neural channel model, an infinite number of comparison channels can be generated on the fly to solve Eq. (5.4), [7] as formalized in Definition 5.3.

Definition 5.3: Location estimator

Let $\mathcal{S} \subset \mathbb{R}^2$ be the set of locations covered by the considered scene. Let $\mathbf{x} \in \mathcal{S}$ be the UE unknown location and $\mathbf{H}(\mathbf{x}) \in \mathbb{C}^{N_a \times N_s}$ be the associated uplink channel matrix. An estimate of $\mathbf{x}$ is given by:

$$\hat{\mathbf{x}}\left(\mathbf{H}(\mathbf{x}) \,|\boldsymbol{\theta}, \mathcal{S}\right) = \underset{\tilde{\mathbf{x}} \in \mathcal{S}}{\arg\min} \left\|\mathbf{H}(\mathbf{x}) - f_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\right\|_{\mathsf{F}}. \tag{5.12}$$

Remark 5.5. *Eq.* (5.12) *is obtained by considering the following similarity measure in Eq.* (5.4)*:*

$$\operatorname{sim}\left(\mathbf{H}(\mathbf{x}_m), \mathbf{H}(\mathbf{x}_n)\right) = 1 - \left\|\mathbf{H}(\mathbf{x}_m) - \mathbf{H}(\mathbf{x}_n)\right\|_{\mathsf{F}}. \tag{5.13}$$

This similarity measure is optimal when one-valued, as it requires the channel coefficients to be exactly equal across all antennas and subcarriers. Its injectivity properties and their

6. See Table 4.1 in Chapter 4.
7. Note that, when comparing the approaches in Section 5.3, the amount of data required to train the generative neural model is taken into account.

implications will be discussed in Section 5.2.3. Since the Frobenius distance component of this similarity measure is central to the subsequent analysis, it is referred to as the phase-sensitive (PS) distance and defined as follows:

$$\mu_{\mathsf{PS}}\left(\mathbf{H}\left(\mathbf{x}\right), \tilde{\mathbf{x}}|\boldsymbol{\theta}\right) = \left\|\mathbf{H}\left(\mathbf{x}\right) - f_{\boldsymbol{\theta}}\left(\tilde{\mathbf{x}}\right)\right\|_{\mathsf{F}}. \tag{5.14}$$

Given the ability of $f_{\boldsymbol{\theta}}$ to effectively learn the location-to-channel mapping, the theoretical error performance of the proposed method can be established, as formalized in Theorem 5.1.

Theorem 5.1: Asymptotic estimation performance

Let $\mathcal{S} \subset \mathbb{R}^2$ be the set of locations covered by the considered scene. Let $\mathbf{x} \in \mathcal{S}$ be the UE unknown location, and $\hat{\mathbf{x}}\left(\mathbf{H}\left(\mathbf{x}\right)|\boldsymbol{\theta}, \mathcal{S}\right)$ be the solution of Eq. (5.12). Let $\epsilon_{f_{\boldsymbol{\theta}}} \in \mathbb{R}^+$ be the model estimation error bound defined in Definition 5.2. Assuming that Eq. (5.13) satisfies Definition 5.1, it follows that:

$$\sup_{\mathbf{x}\in\mathcal{S}} \left\|\mathbf{x} - \hat{\mathbf{x}}\left(\mathbf{H}\left(\mathbf{x}\right)|\boldsymbol{\theta}, \mathcal{S}\right)\right\|_2 \underset{\epsilon_{f_{\boldsymbol{\theta}}}\to 0}{\longrightarrow} 0. \tag{5.15}$$

Proof. See Appendix E.1. ■

Theorem 5.1 demonstrates that the location estimator presented in Definition 5.3 achieves perfect asymptotic accuracy, while maintaining finite memory requirements. Indeed, channel coefficients can be evaluated on-the-fly using $f_{\boldsymbol{\theta}}$, which only requires storing $|\boldsymbol{\theta}|$ coefficients. However, the proposed method requires inferring the channel at every location in $\mathcal{S}$, which is uncountable. To address this, the scene location space is discretized using a finite grid $\mathcal{G} \subset \mathcal{S} \subset \mathbb{R}^2$, leading to the following localization error bound.

Theorem 5.2: Grid-based estimation lower bound

Let $\mathcal{G} \subset \mathcal{S} \subset \mathbb{R}^2$ be the discretized location grid. Let $\mathbf{x} \in \mathcal{S}$ be the UE unknown location. Assuming that the true locations are uniformly distributed within $\mathcal{S}$, and that the grid is uniform with inter-element spacing $\nu \in \mathbb{R}^{+,*}$ yields:

$$\mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}}\left[\left\|\mathbf{x} - \hat{\mathbf{x}}\left(\mathbf{H}\left(\mathbf{x}\right)|\boldsymbol{\theta}, \mathcal{G}\right)\right\|_2\right] \geq \nu\delta, \tag{5.16}$$

with $\delta = \frac{1}{6}\left(\sqrt{2} + \ln\left(\sqrt{2}+1\right)\right) \simeq 0.38$.

Proof. See Appendix E.2. ■

Remark 5.6. *Theorem 5.2 states that the localization accuracy is lower-bounded by the grid inter-element spacing. This is unsurprising as solving Eq.* (5.12) *on the grid* $\mathcal{G}$ *is equivalent to performing a grid search, inherently constraining the localization accuracy to the grid resolution.*

To overcome this grid limitation, an off-grid approach is introduced: after the initial grid search, a gradient-based refinement with N_∇ steps is applied to enhance localization accuracy, as formalized in Eq. (5.17). At step k, the update is expressed as:

$$\hat{\mathbf{x}}^{(k+1)} \leftarrow \hat{\mathbf{x}}^{(k)} - \alpha \nabla_{\tilde{\mathbf{x}}} \mu_{\mathsf{PS}} \left(\mathbf{H}\left(\mathbf{x}\right), \tilde{\mathbf{x}} | \boldsymbol{\theta} \right) |_{\hat{\mathbf{x}}^{(k)}}, \tag{5.17}$$

where $\alpha \in \mathbb{R}^+$ controls the gradient contribution to the update, $\hat{\mathbf{x}}^{(k)}$ is the location estimate at step k, with $\hat{\mathbf{x}}^{(0)} = \hat{\mathbf{x}}\left(\mathbf{H}\left(\mathbf{x}\right) | \boldsymbol{\theta}, \mathcal{G}\right)$. As for all gradient-based methods, this localization refinement is prone to local minima error in the PS distance used as a loss function in Eq. (5.17). A method to avoid local minima and increase performance is presented in Section 5.2.4.

5.2.3 About the channel function injectivity

As discussed in Section 5.1.3, ensuring that the channel function is injective wrt. the similarity measure defined in Eq. (5.13) is essential for the uniqueness of the location estimate, and thus for the accuracy of the proposed method. It can be demonstrated that the channel function is indeed injective wrt. the Frobenius-based similarity function by deriving the global minimum of the PS distance. The subsequent theoretical contributions can be summarized as follows:

- Theorem 5.3 establishes a condition for reaching the global minimum of the Frobenius distance introduced in Eq. (5.14).

- Corollary 5.1 demonstrates that the channel function is injective wrt. the similarity measure defined in Eq. (5.13), based on the results of Theorem 5.3.

- Corollary 5.2 builds on Theorem 5.3, considering a dominant propagation path to define the distance between two consecutive minima in the Frobenius distance.

Theorem 5.3: Global minimum location

For a given location $\mathbf{x} \in \mathcal{S}$, the Frobenius distance component of the similarity measure defined in Eq. (5.14), rewritten as $\|\mathbf{H}(\mathbf{x}) - \mathbf{H}(\mathbf{x}+\boldsymbol{\delta})\|_{\mathsf{F}}$, reaches a global minimum when the displacement vector $\boldsymbol{\delta} \in \mathbb{R}^2$ satisfies:

$$\boldsymbol{\delta} \in \bigcap_{(i,j,l,k)\in\mathcal{L}} \mathcal{C}\left(\mathbf{a}_{l,i} - \mathbf{x}, \|\mathbf{x} - \mathbf{a}_{l,i}\|_2 + k\lambda_j\right). \tag{5.18}$$

with $\mathcal{L} = [\![1, N_a]\!] \times [\![1, N_s]\!] \times [\![1, L_p]\!] \times \mathbb{Z}$, and $\mathbf{a}_{l,i} \in \mathbb{R}^2$ being the antenna locations defined in Remark 5.2.

Proof. See Appendix E.3. ■

Corollary 5.1: Impact on injectivity

In the absence of symmetries in the scene including the antenna array, and considering $N_a > 1$ antennas, the channel function is injective wrt. the similarity measure defined in Eq. (5.13).

Proof. See Appendix E.4. ■

Fig. 5.3 illustrates Theorem 5.3 and Corollary 5.1 by depicting, for each antenna, path, frequency, and integer multiple k, the circles defined in Eq. (5.18). These circles represent local minima, as a displacement vector along one of them nulls the contribution of a specific antenna, frequency, and path in $\|\mathbf{H}(\mathbf{x}) - \mathbf{H}(\mathbf{x}+\boldsymbol{\delta})\|_{\mathsf{F}}$.

Remark 5.7. *As stated by Theorem 5.3, and visualized in Fig. 5.3, the global minimum is attained when all circles intersect, which occurs at* $\mathbf{x}$*, only when* $k = 0$*: this is equivalent to setting* $\boldsymbol{\delta} = \mathbf{0}_{\mathbb{R}^2}$*. This holds only if the scene does not exhibit perfect topological symmetries, when including the antenna array. However, this assumption is weak, as perfect symmetries, up to the scene objects' electromagnetic properties, is unlikely to be found in real-world propagation environments.*

The size of the basins of attraction associated with minima in the Frobenius distance is influenced by the number of antennas and the considered bandwidth. As illustrated in Fig. 5.3 (a), a larger antenna array decreases the tangential extent of the region where the circles intersect. Conversely, the radial size of this region depends on $k(\lambda_{N_s} - \lambda_1), k \in \mathbb{Z}$, which is related to the bandwidth. Theoretically, for a sufficiently high k, or a sufficiently high bandwidth, it is possible to resolve one basin of attraction per frequency. However, in

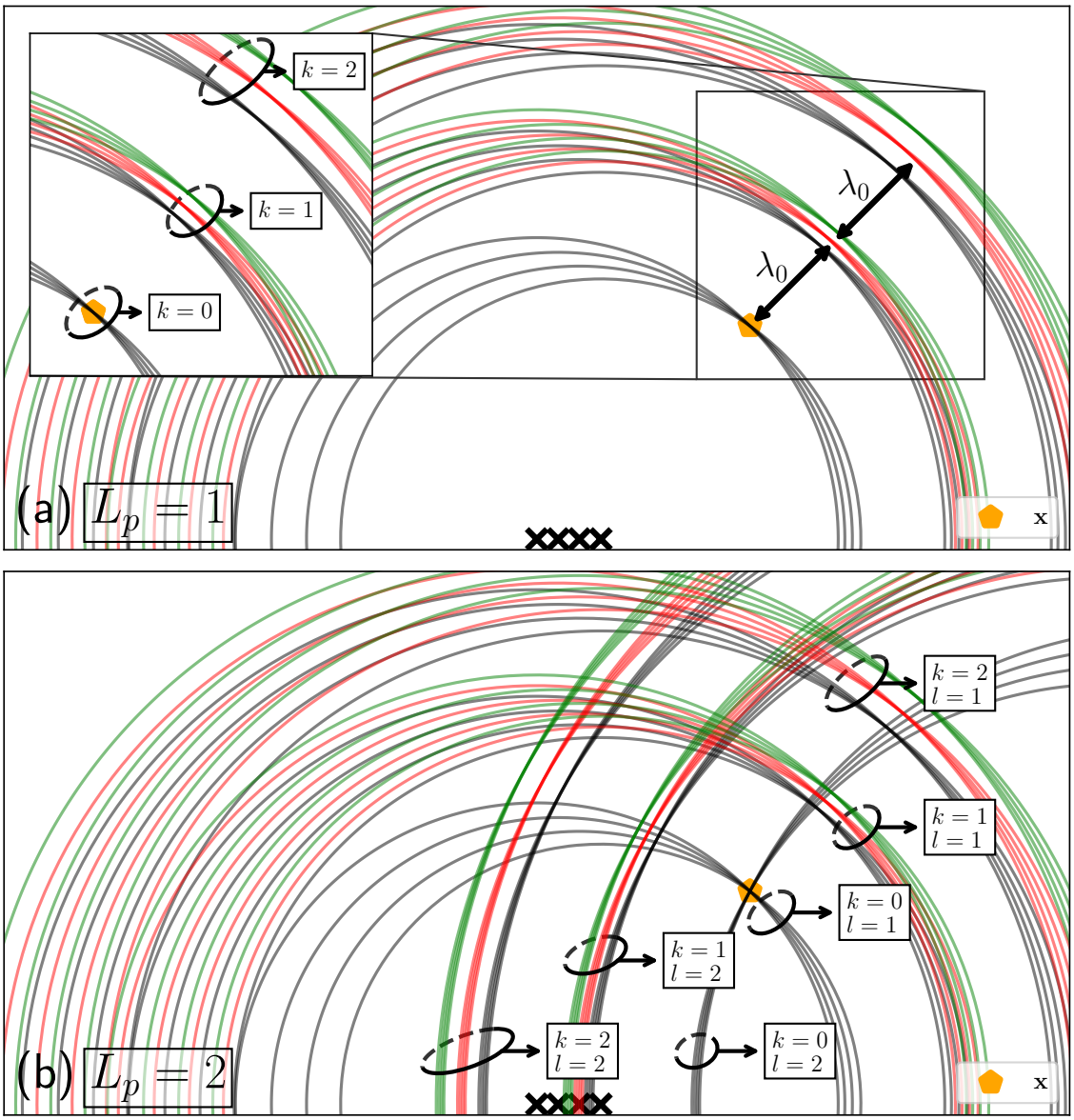


Figure 5.3: Illustration of Theorem 5.3 in a vector space whose origin is $\mathbf{x}$: minimum circles for $N_a = 4$ antennas, $N_s = 3$ frequencies, color-coded as gray/red/green, and $k \in [\![0, 2]\!]$. For $k = 0$, circles originating from all frequencies coincide, as observed in Eq (5.18). The black crosses represent the antenna locations $\mathbf{a}_{1,i}$, $i \in [\![1, N_a]\!]$.

practice, since the wavelength does not significantly vary over the system bandwidth, this distinction remains impractical. This insight clarifies the distance between consecutive minima, as formalized in Corollary 5.2.

Corollary 5.2: Impact on local minima spacing

For a given location $\mathbf{x} \in \mathcal{S}$, in presence of a dominant propagation path, the distance between two consecutive minima in $\|\mathbf{H}(\mathbf{x}) - \mathbf{H}(\mathbf{x} + \boldsymbol{\delta})\|_{\mathsf{F}}$ can be approximated as λ_0, where $\lambda_0 = 1/N_s \sum_{i \in [\![1, N_s]\!]} \lambda_i$.

Proof. See Appendix E.5, and Fig. 5.3 (a) for an illustration. ■

Remark 5.8. *When considering multipath propagation, each propagation path introduces an additional set of circles. In this case, the local minima form a grid-like pattern, as exposed in Fig. 5.3 (b). It increases the complexity of the gradient descent procedure, making convergence to the global minimum more challenging. This is consistent with the well-established fact that localization in the presence of NLoS paths is a hard task.*

5.2.4 Performance and complexity optimization

Hitherto, Theorem 5.2 has established the theoretical limitations of the grid-based approach, Eq. (5.17) has introduced a gradient-based refinement, and Corollary 5.1 has demonstrated the suitability of the proposed similarity measure for this localization task. It is now proposed to optimize the performance of the proposed method, while reducing its computational complexity.

For a given true location, the accuracy of the proposed method depends on two key assumptions. First, the location grid $\mathcal{G}$ must include a point sufficiently close to the true location. Second, the gradient descent process must successfully converge to the global minimum. To satisfy these assumptions, the cardinality of $\mathcal{G}$ needs to be high, which entails significant computational complexity. Additionally, even under this condition, the effectiveness of gradient descent in reaching the global minimum remains uncertain. Indeed, as established in Corollary 5.2, the minima of the PS distance are separated by a distance approximately equal to the central wavelength, which, in conventional communication systems, typically ranges from centimeters to millimeters. This small spacing between local minima underscores the risk of convergence to a local minimum when conducting the gradient-based approach. The proposed enhanced localization method is presented in Algorithm 2; details about performance and complexity optimization are subsequently presented, and Fig. 5.4 illustrates the proposed method and its variants. Note that the *naive* approaches use the large discretized location grid $\mathcal{G}$.

Algorithm 2 Proposed enhanced localization method: Off-grid (PI/PS).

Inputs: Measured channel matrix $\mathbf{H}(\mathbf{x}) \in \mathbb{C}^{N_a \times N_s}$, trained neural model $f_{\boldsymbol{\theta}}$, selected initialization loss μ_{init}, global coarse location grid $\mathcal{G}_{\mathsf{G}}$, local grid size L, inter-element spacing ν, number of circles N_{C}, number of circle points M_{C}.

1: *((a) in Fig. 5.4).* Compute the local grid initialization location by performing a grid search on $\mathcal{G}_{\mathsf{G}}$: $\tilde{\mathbf{x}}_{\mathrm{i}} \leftarrow \hat{\mathbf{x}}\left(\mathbf{H}(\mathbf{x})\,|\boldsymbol{\theta}, \mathcal{G}_{\mathsf{G}}\right)$, with:

$$\hat{\mathbf{x}}\left(\mathbf{H}(\mathbf{x})\,|\boldsymbol{\theta}, \mathcal{G}_{\mathsf{G}}\right) = \underset{\tilde{\mathbf{x}} \in \mathcal{G}_{\mathsf{G}}}{\arg\min}\, \mu_{\mathsf{init}}\left(\mathbf{H}(\mathbf{x}), \tilde{\mathbf{x}}|\boldsymbol{\theta}\right). \tag{5.19}$$

2: Construct the local grid $\mathcal{G}_{\mathsf{L}}$: size L by L, centered at $\tilde{\mathbf{x}}_{\mathrm{i}}$, with inter-element spacing ν in both directions:

$$\mathcal{G}_{\mathsf{L}} = \left(\tilde{x}_{\mathrm{i},0} + \mathbb{Z}_{\nu,L}\right) \times \left(\tilde{x}_{\mathrm{i},1} + \mathbb{Z}_{\nu,L}\right), \tag{5.20}$$

with $\mathbb{Z}_{\nu,L} = \nu\mathbb{Z} \cap [-L/2, L/2]$.

3: *((b) in Fig. 5.4).* Perform a grid search on $\mathcal{G}_{\mathsf{L}}$: $\tilde{\mathbf{x}}_{\mathrm{g}} \leftarrow \hat{\mathbf{x}}\left(\mathbf{H}(\mathbf{x})\,|\boldsymbol{\theta}, \mathcal{G}_{\mathsf{L}}\right)$, with:

$$\hat{\mathbf{x}}\left(\mathbf{H}(\mathbf{x})\,|\boldsymbol{\theta}, \mathcal{G}_{\mathsf{L}}\right) = \underset{\tilde{\mathbf{x}} \in \mathcal{G}_{\mathsf{L}}}{\arg\min}\, \mu_{\mathsf{PS}}\left(\mathbf{H}(\mathbf{x}), \tilde{\mathbf{x}}|\boldsymbol{\theta}\right). \tag{5.21}$$

4: *((c) in Fig. 5.4).* Perform N_{∇} gradient steps from $\tilde{\mathbf{x}}_{\mathrm{g}}$, as defined in Eq. (5.17), to obtain $\tilde{\mathbf{x}}_{\mathrm{gd}}$.

5: Evaluate the Frobenius distance at the obtained location:

$$\gamma_{\mathrm{gd}} = \mu_{\mathsf{PS}}\left(\mathbf{H}(\mathbf{x}), \tilde{\mathbf{x}}_{\mathrm{gd}}|\boldsymbol{\theta}\right). \tag{5.22}$$

6: Sample locations on the circles centered at $\tilde{\mathbf{x}}_{\mathrm{gd}}$, with radii $k\lambda_0, k \in [\![1, N_{\mathsf{C}}]\!]$:

$$\mathcal{G}_{\mathsf{C}} \subset \bigcup_{k \in [\![1, N_{\mathsf{C}}]\!]} \mathcal{C}\left(\tilde{\mathbf{x}}_{\mathrm{gd}}, k\lambda_0\right), |\mathcal{G}_{\mathsf{C}}| = M_{\mathsf{C}}. \tag{5.23}$$

7: *((d) in Fig. 5.4).* Find the minimum Frobenius distance on the circles:

$$\tilde{\mathbf{x}}_{\mathrm{c}}^{\star} = \underset{\tilde{\mathbf{x}}_{\mathrm{c}} \in \mathcal{G}_{\mathsf{C}}}{\arg\min}\, \mu_{\mathsf{PS}}\left(\mathbf{H}(\mathbf{x}), \tilde{\mathbf{x}}_{\mathrm{c}}|\boldsymbol{\theta}\right) \tag{5.24}$$

$$\gamma_{\mathrm{c}}^{\star} = \mu_{\mathsf{PS}}\left(\mathbf{H}(\mathbf{x}), \tilde{\mathbf{x}}_{\mathrm{c}}^{\star}|\boldsymbol{\theta}\right). \tag{5.25}$$

Compare the Frobenius distances:

8: **if** $\gamma_{\mathrm{c}}^{\star} \leq \gamma_{\mathrm{gd}}$ **then**

9: *((e) in Fig. 5.4).* Perform N_{∇} gradient steps from $\tilde{\mathbf{x}}_{\mathrm{c}}^{\star}$, as defined in Eq. (5.17), to obtain $\tilde{\mathbf{x}}_{\mathrm{gd}_2}$.

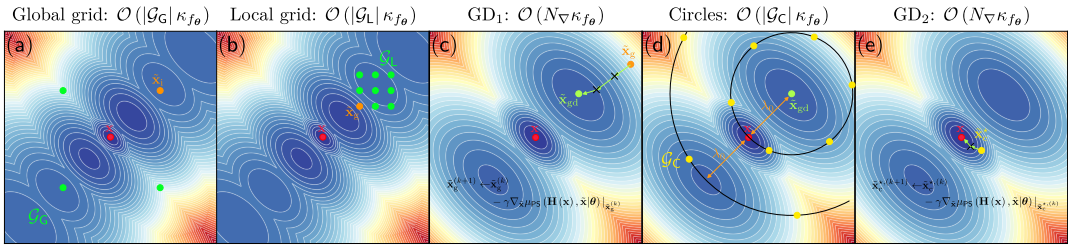
10: Update the estimated location: $\hat{\mathbf{x}} \leftarrow \tilde{\mathbf{x}}_{\mathrm{gd}_2}$.

11: **else**

12: Update the estimated location: $\hat{\mathbf{x}} \leftarrow \tilde{\mathbf{x}}_{\mathrm{gd}}$.

13: **end if**

Output: $\hat{\mathbf{x}}$ (Estimated location)



On-grid (naive)	✓ μ_PS on $\mathcal{G}$	✗	✗	✗	✗
On-grid (PI/PS)	✓ $\mu_\mathsf{PI}/\mu_\mathsf{PS}$	✓ μ_PS	✗	✗	✗
Off-grid (naive)	✓ μ_PS on $\mathcal{G}$	✗	✓	✓	✓
Off-grid (PI/PS)	✓ $\mu_\mathsf{PI}/\mu_\mathsf{PS}$	✓ μ_PS	✓	✓	✓

Figure 5.4: Illustration of the proposed localization scheme steps. (a): a grid-search is performed on the global grid $\mathcal{G}_\mathsf{G}$. (b): a second grid-search is performed on the local grid $\mathcal{G}_\mathsf{L}$, constructed from the obtained location at the previous step. (c): a first gradient-descent procedure is performed, and a local minimum is reached. (d): locations are sampled on circles of center $\tilde{\mathbf{x}}_{\mathrm{gd}}$ and radii $k\lambda_0$. (e): a second gradient-descent procedure is performed from the best location in the sampled circle locations, from a PS distance perspective. The bottom row illustrates the steps and loss functions involved in the proposed method (Off-grid (PI/PS)) and its variants, which are presented in details in Section 5.3.

Performance optimization

The theoretical knowledge of the distance between consecutive minima is incorporated into the gradient-based refinement procedure. Specifically, a dual-gradient-descent scheme is considered: the first gradient descent is performed from $\tilde{\mathbf{x}}_{\mathrm{g}}$, i.e. the location obtained through the on-grid approach. This approach allows to obtain the location estimate $\tilde{\mathbf{x}}_{\mathrm{gd}}$. Subsequently, M_C locations are sampled on N_C circles of center $\tilde{\mathbf{x}}_{\mathrm{gd}}$ and radii corresponding to integer multiples of the theoretical separation distance between minima, i.e. $k\lambda_0, k \in \mathbb{Z}$ (line 6 in Algorithm 2). The minimum Frobenius distance on the sampled circle locations, i.e. $\gamma_{\mathrm{c}}^\star$, associated to location $\tilde{\mathbf{x}}_{\mathrm{c}}^\star$, is then compared to the value obtained at the location estimated through the initial gradient-descent procedure. If it is lower, it indicates the presence of a better minimum, suggesting the possibility of refining the location estimate towards the global minimum. In that case, a second gradient-descent procedure is performed from $\tilde{\mathbf{x}}_{\mathrm{c}}^\star$ to ensure convergence to the lowest point within the basin of attraction.

Complexity optimization

To mitigate the computational complexity of the proposed method, a bi-level grid approach is proposed. Initially, an exhaustive search is performed on a coarse grid, denoted as $\mathcal{G}_\mathsf{G} \subset \mathcal{S}$, to determine an initial location estimate $\tilde{\mathbf{x}}_{\mathrm{i}}$. This *global* grid is constructed based solely on the topological knowledge of the scene $\mathcal{S}$, which is used to define the

Table 5.1: Complexity comparison of the proposed approaches.

	On-grid (naive)	Off-grid (naive)	Off-grid (PI/PS)
Time complexity	$\mathcal{O}\left(\lvert\mathcal{G}\rvert\,\kappa_{f_{\boldsymbol{\theta}}}\right)$	$\mathcal{O}\left(\left(\lvert\mathcal{G}\rvert+\lvert\mathcal{G}_{\mathsf{C}}\rvert+2N_{\nabla}\right)\kappa_{f_{\boldsymbol{\theta}}}\right)$	$\mathcal{O}\left(\left(\lvert\mathcal{G}_{\mathsf{G}}\rvert+\lvert\mathcal{G}_{\mathsf{L}}\rvert + \lvert\mathcal{G}_{\mathsf{C}}\rvert+2N_{\nabla}\right)\kappa_{f_{\boldsymbol{\theta}}}\right)$
Forward passes nb.	218k	219.2k	11.2k

grid boundaries. Subsequently, a finer *local* grid, $\mathcal{G}_{\mathsf{L}} \subset \mathcal{S}$, is generated around $\tilde{\mathbf{x}}_{\mathrm{i}}$, and a refined location estimate, $\tilde{\mathbf{x}}_{\mathrm{g}}$, is obtained through a grid search over $\mathcal{G}_{\mathsf{L}}$. Finally, a gradient-descent optimization process is applied starting from this refined location estimate. The loss function used to obtain the initialization location, as defined in Eq. (5.19), can either be the Frobenius distance loss presented in Eq. (5.14) or the phase-insensitive (PI) distance introduced in [LeMag21a], which is defined as follows [8]:

$$\mu_{\mathsf{PI}}\left(\mathbf{H}\left(\mathbf{x}\right),\tilde{\mathbf{x}}|\boldsymbol{\theta}\right)=\sqrt{2-2\frac{\left|\mathrm{Tr}\left(\mathbf{H}\left(\mathbf{x}\right)^{\mathsf{H}}f_{\boldsymbol{\theta}}\left(\tilde{\mathbf{x}}\right)\right)\right|}{\left\|\mathbf{H}\left(\mathbf{x}\right)\right\|_{\mathsf{F}}\left\|f_{\boldsymbol{\theta}}\left(\tilde{\mathbf{x}}\right)\right\|_{\mathsf{F}}}}. \tag{5.26}$$

This PI distance does not distinguish maxima from minima due to the modulus operator in its definition. While this characteristic limits its applicability for achieving precise localization, it enhances the reliability of obtaining an estimate sufficiently close to the global minimum. Indeed, this inability to distinguish between minima and maxima results in a more gradual minimum profile, in contrast to the highly localized nature of the PS distance minima. The respective behaviors of the PS and PI distances for this localization problem will be illustrated in Section 5.3.

A complexity comparison of the proposed approaches is presented in Table 5.1, where $\kappa_{f_{\boldsymbol{\theta}}}$ represents the network forward pass complexity. The *on-grid* approach refers to the localization without gradient-based refinement, whereas the *off-grid* approach considers it. The PI/PS notation indicates which loss function is used for the grid-search on $\mathcal{G}_{\mathsf{G}}$. As the bi-level grid approach allows to obtain a local grid center estimate that is close to the global minimum, it is possible to consider a low cardinality in $\mathcal{G}_{\mathsf{L}}$, so that:

$$\lvert\mathcal{G}_{\mathsf{G}}\rvert+\lvert\mathcal{G}_{\mathsf{L}}\rvert+\lvert\mathcal{G}_{\mathsf{C}}\rvert+2N_{\nabla}\ll\lvert\mathcal{G}\rvert\,. \tag{5.27}$$

Specifically, considering $\lvert\mathcal{G}\rvert = 218.10^3$, i.e. uniform grid on a 100 m^2 square scene, with $\lambda_0/4$ inter-element spacing, $\lvert\mathcal{G}_{\mathsf{G}}\rvert = 1.10^3$, $\lvert\mathcal{G}_{\mathsf{L}}\rvert = 9.10^3$, $\lvert\mathcal{G}_{\mathsf{C}}\rvert = 1.10^3$, and $N_{\nabla} = 100$ yields the second line in Table 5.1, which clearly highlights the overall complexity improvement obtained with the proposed approach.

8. This distance corresponds to the one presented in Chapter 8, in Eq. (8.11), expressed here in its matrix formulation.

Remark 5.9. *The proposed localization method builds on the domain knowledge formalized in (H1). As detailed in Chapter 4, the neural architecture used to learn the location-to-channel mapping is derived from a propagation channel model based on (H1). Moreover, the improved localization method leverages a circle-based procedure, itself grounded in the theoretical characterization of the distance between consecutive minima of the channel similarity function, as established in Corollary 5.2 using the same propagation channel model. Both components ultimately stem from the channel model originating from (H1).*

5.3 Experiments

In this section, the localization performance of the proposed estimator is evaluated against traditional fingerprinting baselines such as k-NN, on realistic synthetic data in both indoor and outdoor scenes.

5.3.1 Evaluation framework

System parameters

For all upcoming experiments, $N_s = 64$ frequencies are considered, with a central frequency $f_0 = 3.5$ GHz and a bandwidth of 20 MHz. The associated central wavelength is $\lambda_0 \simeq 8.57$ cm. The BS is equipped with a uniform linear array composed of $N_a = 64$ antennas with half central wavelength spacing.

Scene characteristics

Three distinct scenes are considered, as represented in Fig. 5.5: an outdoor urban environment in the *Etoile* neighborhood of Paris, denoted as S_1, and two distinct indoor environments with metallic obstacles, denoted as S_2 and S_3. The primary distinction between S_2 and S_3 is the higher proportion of locations without LoS paths in S_3 (around 50%), making it representative of a more complex radio environment. Additionally, in S_1, the antenna array is located close to the UE locations, while it is far in S_2 and S_3. These contrasting settings enable the evaluation of the proposed localization method in scenarios representative of outdoor cellular deployments and indoor industrial environments.

Location generation

Each scene consists of a $L_{\mathcal{S},x}L_{\mathcal{S},y}$ m^2 plane considered as the location space $\mathcal{S}$. Within this location space, training, test, and evaluation locations are sampled. The training locations, used to train the $f_{\boldsymbol{\theta}}$ model, and denoted as $\mathcal{G}_{\mathsf{train}}$, are uniformly sampled as follows:

$$\forall \mathbf{x}_i \in \mathcal{G}_{\mathsf{train}}, \mathbf{x}_i \sim \mathcal{U}\left([-L_{\mathcal{S},x}/2, L_{\mathcal{S},x}/2] \times [-L_{\mathcal{S},y}/2, L_{\mathcal{S},y}/2]\right). \tag{5.28}$$

Test locations, denoted as $\mathcal{G}_{\mathsf{test}}$, are used to validate the mapping-learning capabilities of $f_{\boldsymbol{\theta}}$. They form a uniform grid with an inter-element spacing of $\lambda_0/4$: this yields a train/test

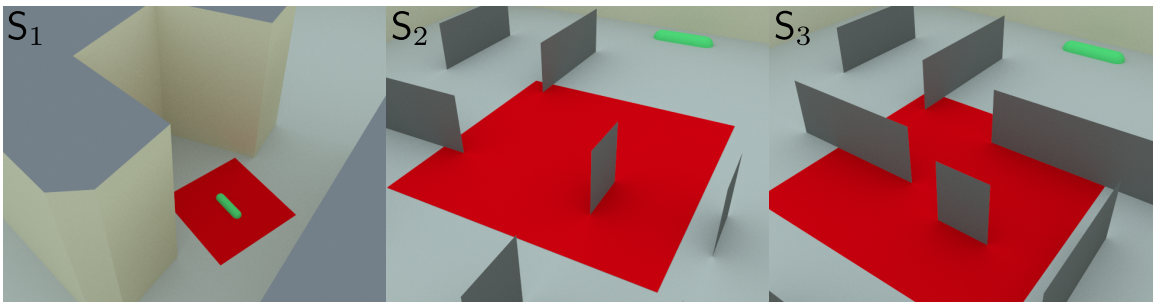


Figure 5.5: 3D view of the considered scenes: the red plane represents the possible training/test/evaluation locations and the green spheres represent the BS antennas. In S_2 and S_3, gray planes represent metallic obstacles. For S_1, only the locations in front of the BS are considered during the localization performance evaluation.

ratio of around 7%. Evaluation locations, collected in $\mathcal{G}_{\mathsf{loc}}$, are uniformly sampled within $\mathcal{S}$, following Eq. (5.28), and are specifically used to assess the performance of the proposed localization method. Importantly, $\mathcal{G}_{\mathsf{train}} \cap \mathcal{G}_{\mathsf{test}} = \varnothing$, $\mathcal{G}_{\mathsf{train}} \cap \mathcal{G}_{\mathsf{loc}} = \varnothing$, and $\mathcal{G}_{\mathsf{test}} \cap \mathcal{G}_{\mathsf{loc}} = \varnothing$, ensuring that localization performance is evaluated on locations independent of those used for training and testing. Note that for S_1, $L_{\mathcal{S},x} = 10$ m and $L_{\mathcal{S},y} = 5$ m, while $L_{\mathcal{S},x} = L_{\mathcal{S},y} = 10$ m for S_2 and S_3. Additionally, for all scenes, the training location density is set to 175 locs./m$^2 \simeq 1.3$ locs./λ_0^2.

Remark 5.10. *Results from our prior study [Cha+25a], i.e. Fig. 4.8 in Chapter 4, demonstrated effective mapping learning at a training location density of* 0.5 *locs/*λ_0^2*, suggesting that satisfactory localization performance can be expected at lower densities.*

Channel generation

The datasets are obtained through the radio-propagation digital twin of the Sionna library [Hoy+22]. Since ray-tracing methods rely on fundamental electromagnetic principles, the resulting channels, both in indoor and outdoor scenes, can be considered realistic. For each antenna and location in the scene, ray-tracing techniques identify the propagation paths and compute the corresponding frequency-domain channel coefficients. The number of propagation paths depends on the location, with up to two consecutive reflections considered, while diffraction is neglected. The complete dataset consists of the locations and associated channel matrices, e.g. the training dataset is defined as:

$$\mathcal{D}_{\mathsf{tr}} = \{\mathbf{x}_i \in \mathcal{G}_{\mathsf{train}}, \mathbf{H}(\mathbf{x}_i) \in \mathcal{H}_{\mathsf{train}}\}_{i=1}^{N_{\mathsf{train}}}, \tag{5.29}$$

where $\mathcal{H}_{\mathsf{train}}$ consists of the computed channel matrices at the training locations, and $N_{\mathsf{train}} = |\mathcal{G}_{\mathsf{train}}|$ is the number of training locations.

Evaluation metrics

Let $\mathbf{x}$ denote the true UE location and $\hat{\mathbf{x}}$ its estimate. The performance of the localization methods is evaluated through statistical analysis of the localization error $\|\mathbf{x} - \hat{\mathbf{x}}\|_2$ over the considered scene, using metrics such as the median, 10% and 90% quantiles.

Baselines

It is proposed to consider the weighted k-NN method on the training dataset as a fingerprinting baseline. Specifically, for a given channel $\mathbf{H}(\mathbf{x})$, the k-best locations are computed as:

$$\{\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_k\} = \underset{\tilde{\mathbf{x}} \in \mathcal{G}_{\mathsf{train}}}{\arg\min}_k \frac{\|\mathbf{H}(\mathbf{x}) - \mathbf{H}(\tilde{\mathbf{x}})\|_{\mathsf{F}}^2}{\|\mathbf{H}(\mathbf{x})\|_{\mathsf{F}}^2}, \tag{5.30}$$

and the associated weights are, $\forall i \in [\![1, k]\!]$:

$$\alpha_i = \left(\frac{\|\mathbf{H}(\mathbf{x}) - \mathbf{H}(\tilde{\mathbf{x}}_i)\|_{\mathsf{F}}^2}{\|\mathbf{H}(\mathbf{x})\|_{\mathsf{F}}^2} \right)^{-1}, \tag{5.31}$$

which are then normalized as $\tilde{\alpha}_i = \alpha_i / \sum_{i=1}^{k} \alpha_i$. The estimated location through k-NN is finally computed as:

$$\hat{\mathbf{x}} = \sum_{i=1}^{k} \tilde{\alpha}_i \tilde{\mathbf{x}}_i. \tag{5.32}$$

Considering this fingerprinting baseline on the training dataset, used to train the neural generative model, ensures a fair comparison with the proposed method, as both approaches use exactly the same data to generate a location estimate.

Additionally, a MLP fingerprinting baseline is also considered. This MLP has four complex hidden layers of sizes 4096, 2048, 1024, and 512, takes channel matrices as input, and outputs a location estimate. It comprises approximately 22 million learnable parameters.

5.3.2 Experimental results

Unless stated otherwise, in each scene, the following grid cardinalities are considered: $|\mathcal{G}_{\mathsf{G}}| = 1.10^3$, $|\mathcal{G}_{\mathsf{L}}| = 140.10^3$ for the PS init. method on the S_2 and S_3 scenes, and $|\mathcal{G}_{\mathsf{L}}| = 9.10^3$ for both the PS init. method on the S_1 scene, and the PI init. method on all scenes. This cardinality difference in complex radio environments originates from the need for a larger coverage area in the PS case, which is attributed to the PS distance yielding a less accurate initial location estimate than the PI distance in such scenarios. Finally, $N_{\nabla} = 100$ gradient descent steps are considered.

PI/PS distance behavior

Fig. 5.6 illustrates the behavior of the PI and PS distances in the S_1 scene for a given true location $\mathbf{x}$. It can be observed that the PS distance exhibits highly localized minima, which hinders the reliability of the initialization location estimate through the grid search on the global grid. In contrast, the PI distance exhibits a smoother, more gradual loss profile, with very high loss values anywhere except within a broad neighborhood in the true location direction, illustrated by the yellow area in the zoomed-out view of Fig. 5.6. This facilitates the identification of an accurate initial location estimate through the global grid search. The closer spacing of minima in the PI distance, compared to the PS distance, can be attributed to its inability to distinguish maxima from minima, as previously discussed.

Localization performance

Fig. 5.7 presents the performance of each approach by considering the boxplots of the localization error on 1.10^4 independent locations. The low/high bars of the boxplots respectively represent the 10% and 90% quantiles, the central orange line represents the median, and the box boundaries correspond to the first and third quartiles. The different methods are as follows:

- MLP and 3-NN: fingerprinting baselines.
- On-grid (PI/PS): proposed method using the PI or PS loss in the exhaustive search on $\mathcal{G}_{\mathsf{G}}$, but without the gradient-descent refinement.
- Off-grid (PS w/o circle): proposed method using the PS loss in the exhaustive search on $\mathcal{G}_{\mathsf{G}}$, with gradient-descent, but without the circles approach (see Fig. 5.4 (d)).
- Off-grid (PI/PS): proposed method using the PI or PS loss in the exhaustive search on $\mathcal{G}_{\mathsf{G}}$, with both gradient-descent and circles approaches.

The proposed off-grid (PI) method consistently outperforms the fingerprinting baselines across all scenes, achieving an improvement of two to three orders of magnitude in the median localization performance. Notably, its median value reaches 0.01 cm in S_1, and approximately 0.06 cm in S_3, demonstrating its strong performance throughout the entire scene, and across challenging radio environments. It is worth noting that these values are well below the central wavelength, $\lambda_0 \simeq 8.57$ cm, highlighting the sub-wavelength accuracy of the proposed method. As anticipated, the off-grid approach significantly improves localization accuracy by eliminating the constraint of a fixed grid. Moreover, the median error values for the on-grid PS approach are approximately on the order of the central wavelength, highlighting the high sensitivity of the PS distance to local minima. In contrast, the PI distance demonstrates greater robustness to this issue, corroborating the insights from Eq. (5.26) and Fig. 5.6. For both initialization methods, the circle-based approach effectively mitigates this limitation, and enhances localization resolution. The degraded performance of the proposed method in the S_2 and S_3 scenes, compared to S_1,

can be attributed to the increased complexity of the radio environments simulated in S_2 and S_3. The presence of strong NLoS paths, and the complete absence of LoS paths for around 50% of the locations in S_3 results in a much more complex loss landscape, making convergence to the global minimum more challenging. In this scenario, the circle-based approach, which is designed to mitigate the local minima issue, reaches its limit as the PS distance function exhibits increasingly chaotic minima. This can be seen with the superior localization performance of the off-grid (PI) approach on the S_3 scene when considering only locations with LoS paths (denoted as w. LoS only in Fig. 5.7), as compared to locations with only NLoS paths (denoted as w. NLoS only in Fig. 5.7). This chaotic minima behavior is illustrated in Fig. 5.8 with the loss landscape over the S_1 and S_3 scenes.

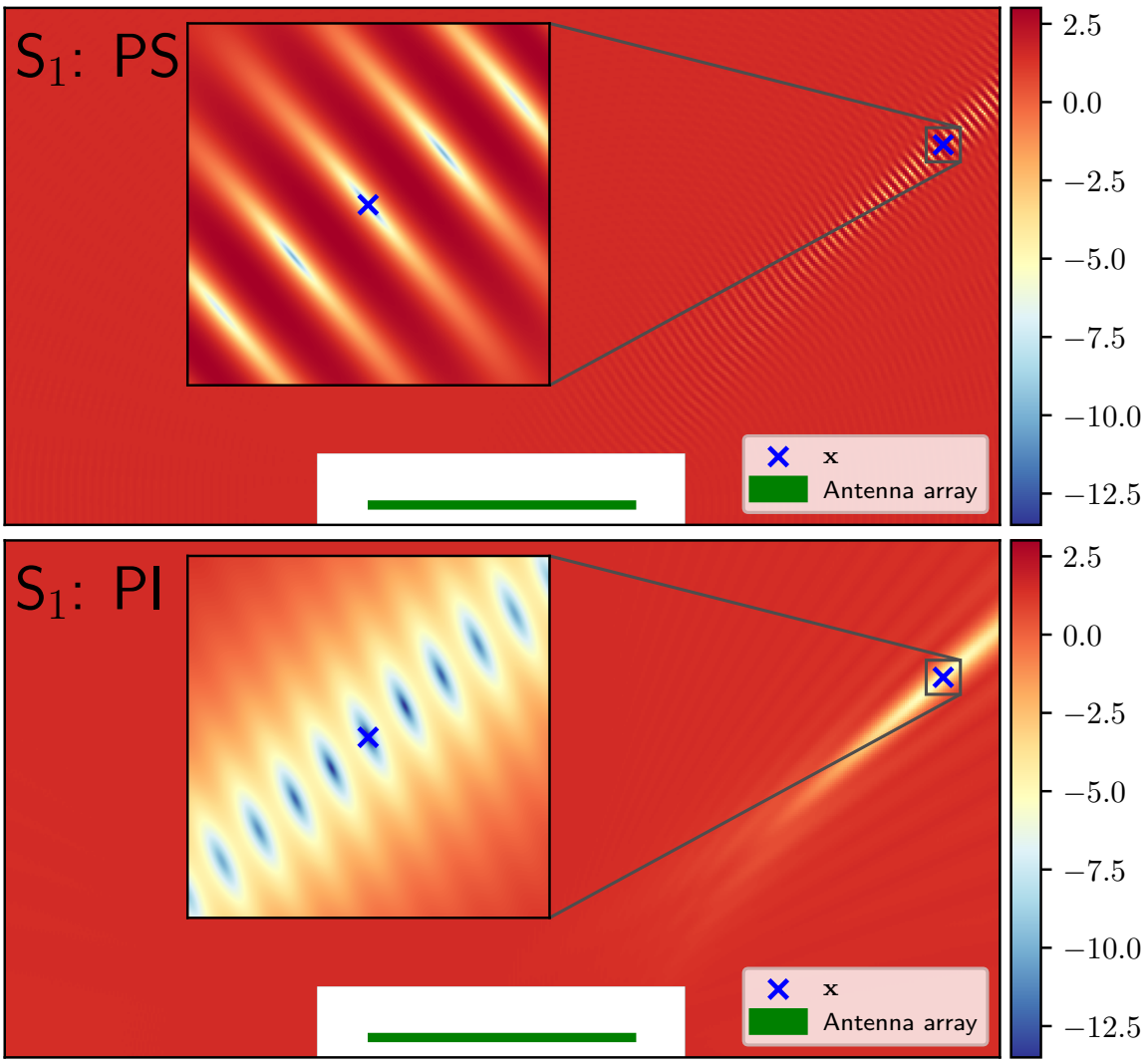


Figure 5.6: Comparison of $\mu_{\mathsf{PS}}\left(\mathbf{H}\left(\mathbf{x}\right), \tilde{\mathbf{x}}|\boldsymbol{\theta}\right)$ and $\mu_{\mathsf{PI}}\left(\mathbf{H}\left(\mathbf{x}\right), \tilde{\mathbf{x}}|\boldsymbol{\theta}\right)$, in dB, for a given true location $\mathbf{x}$ in the S_1 scene.

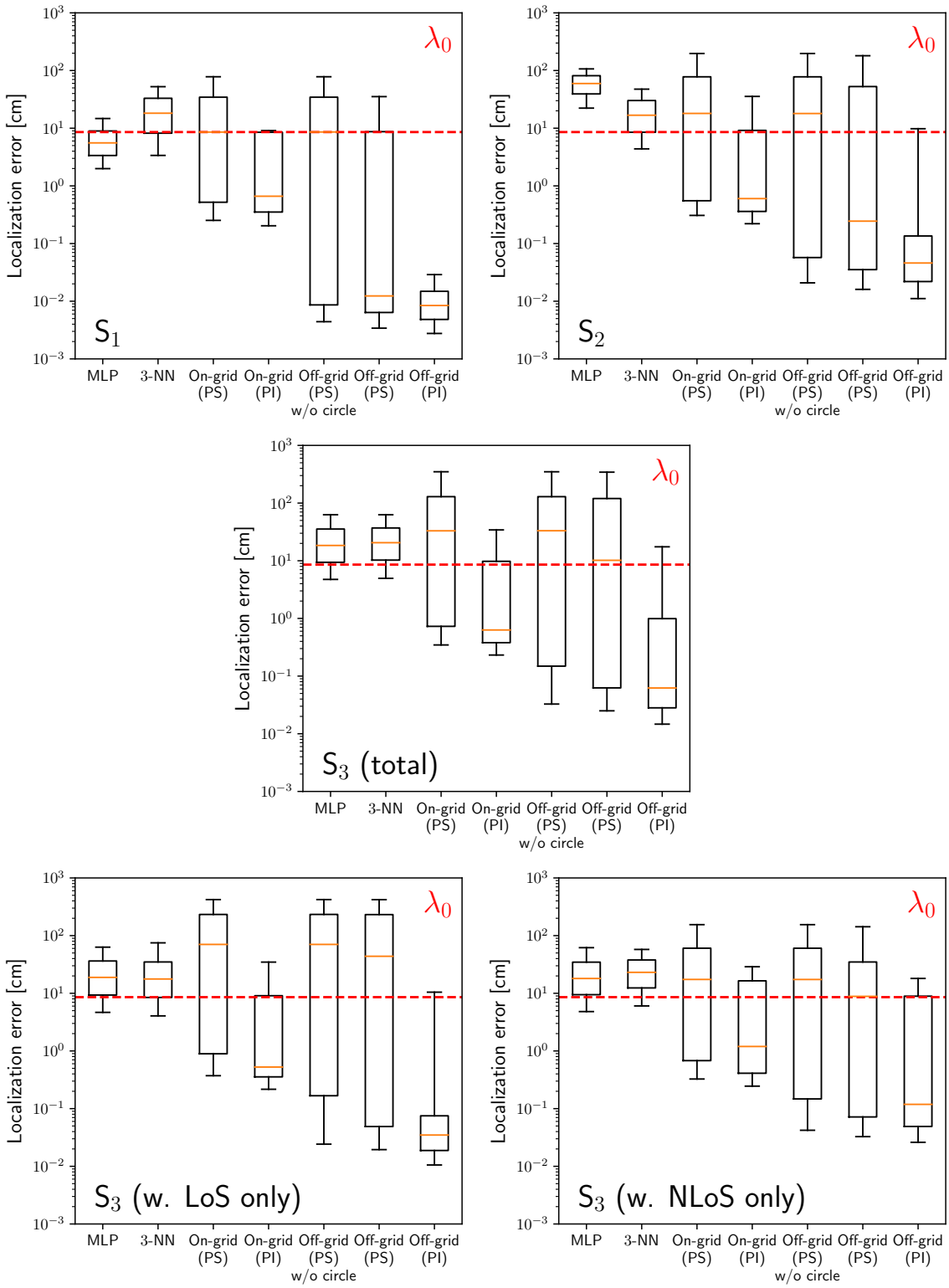


Figure 5.7: Statistics on the localization error (in cm) across the S_1, S_2 and S_3 scenes. The red horizontal line represents the central wavelength λ_0.

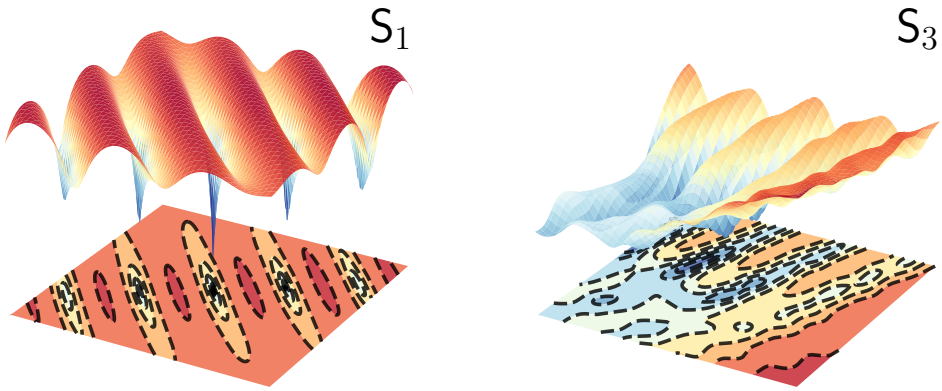


Figure 5.8: PS distance landscapes for the S_1 and S_3 scenes. It is observed that the presence of strong NLoS components is S_3 yield a more complex loss landscape.

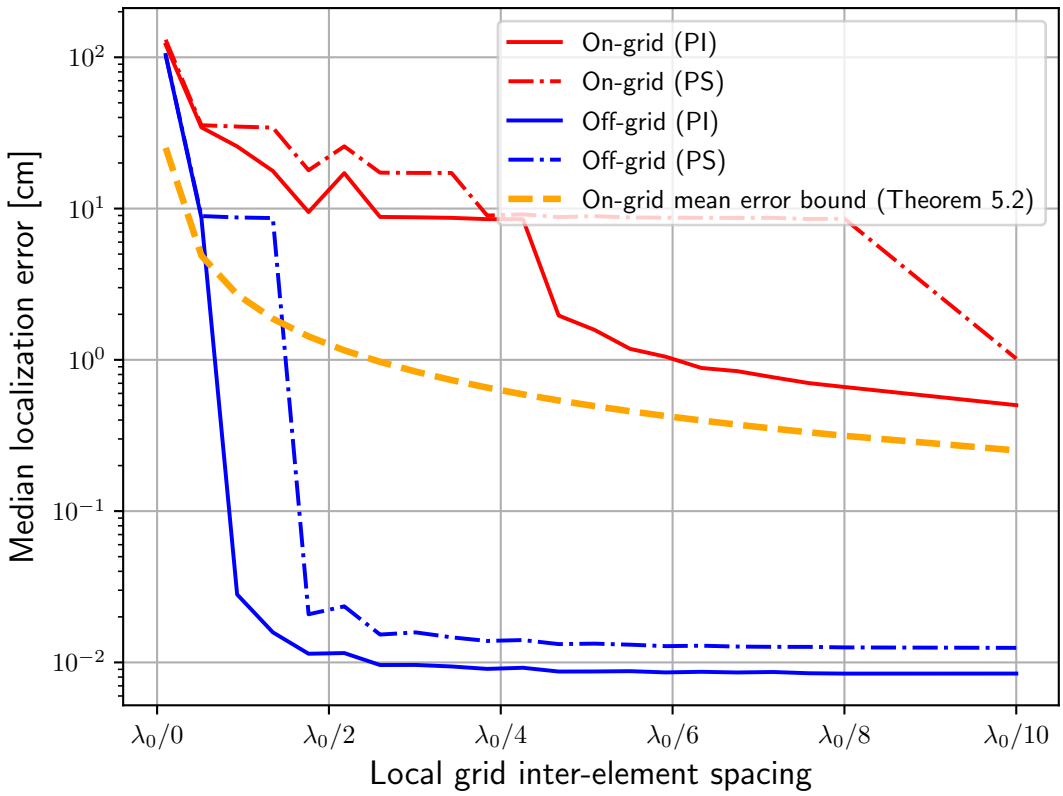


Figure 5.9: Localization performance evolution wrt. $\mathcal{G}_{\mathsf{L}}$ inter-element spacing, over the S_1 scene.

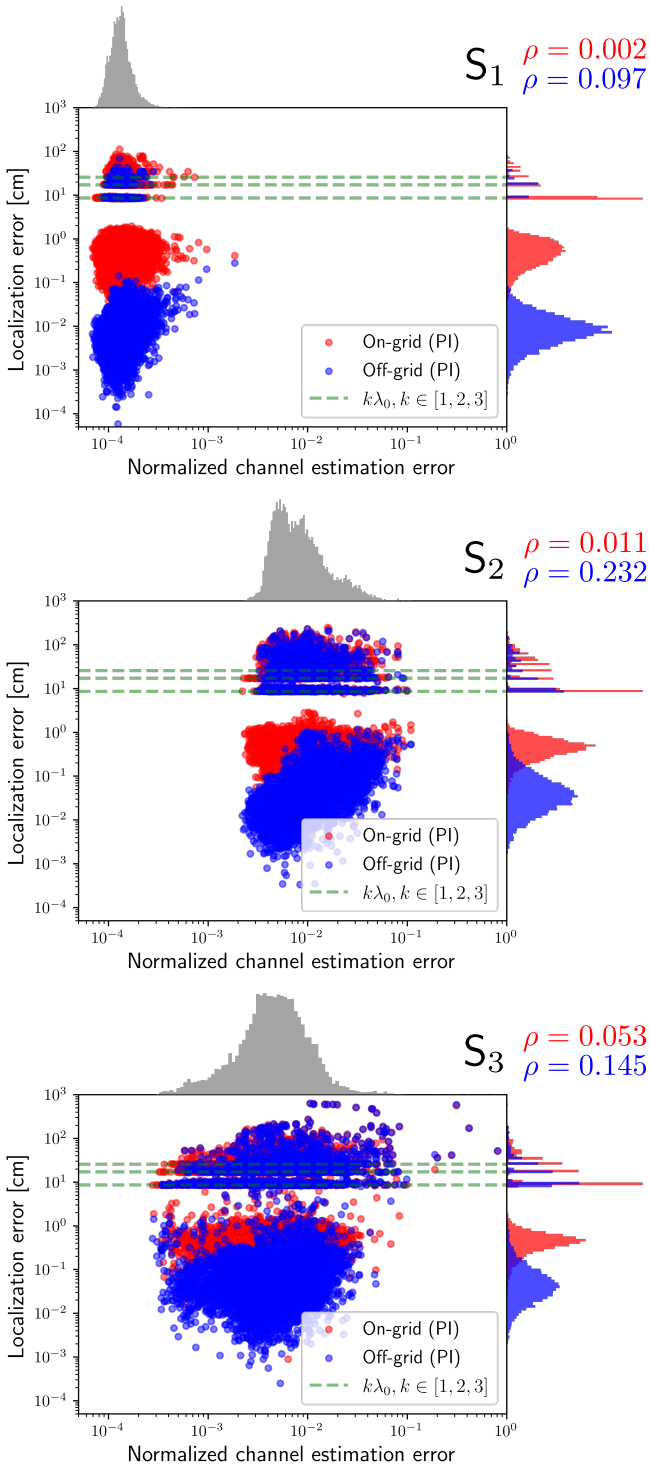


Figure 5.10: 2D histogram of the location error wrt. the model error, across the S_1, S_2 and S_3 scenes. A zero-valued Pearson's correlation coefficient ρ indicates that the location and channel estimation errors are uncorrelated. Green dashed lines represent integer multiples of the central wavelength λ_0.

Local grid spacing performance impact

Fig. 5.9 illustrates the median localization error performance as a function of the inter-element spacing in $\mathcal{G}_{\mathsf{L}}$, over the S_1 scene. It is observed that the on-grid approach requires a very dense grid to approach the best-case bound defined in Theorem 5.2. However, as anticipated, the off-grid approach surpasses this on-grid best-case performance bound, even with a relatively coarse grid. This demonstrates the efficiency of the off-grid approach in delivering accurate localization estimates, while maintaining a controlled computational complexity. This is enabled by the ability of the $f_{\boldsymbol{\theta}}$ model to serve as a generative neural channel model with low estimation error.

Relationship between model and localization performance

In Fig. 5.10, each point in the bi-dimensional histograms corresponds, for a given true location, to the pair formed by the localization error and the channel estimation error, i.e. NMSE, at that true location, with marginals displayed on the right and top axes. In the S_1 scene, the near-zero Pearson's correlation coefficient indicates that the localization error is largely uncorrelated with the model error, whereas a noticeable correlation appears in the S_2 and S_3 scenes. This is likely due to the higher model NMSE in S_2 and S_3 (around -10 dB), compared to S_1 (around -30dB). Nevertheless, even with high model NMSE, localization performance remains satisfactory. This implies that even when the reference channel, i.e. the channel estimated by the network at the true location, is imperfectly reconstructed, the resulting location estimate remains highly accurate, highlighting the robustness of the proposed method to noisy channels during inference. Additionally, it can be observed that in S_2 and S_3, the off-grid approach benefits only at locations where the model error remains within acceptable limits. This behavior is consistent across all locations in S_1, where the model error is uniformly low due to the less challenging nature of the scene. Furthermore, localization errors tend to accumulate at multiples of the central wavelength: this is an expected effect of the local minima in the PS distance, as discussed in Corollary 5.2. The off-grid approach, which incorporates the circles method, partially mitigates this local minima issue: owing to its ability to escape local minima through the circles sampling strategy, the proposed method can be viewed as a model-based annealing approach.

Computational complexity

Table 5.2 reports the empirical inference computational complexity of the proposed methods and baselines over the S_2 scene. The average inference time for each method is computed over 1.10^4 independent locations, with all computations performed on the same NVIDIA A40 GPU. The on-grid (naive) approach employs a uniform grid across the scene with an inter-element spacing of $\lambda_0/4$. The off-grid (naive) approach extends it by incorporating the proposed gradient descent and circle-based methods, but lacks the computational complexity optimization provided by the bi-level grid approach. Lastly, the

Table 5.2: Average inference time over the S_2 scene.

	Baselines		On-grid	Off-grid		
	MLP	3-NN	Naive	Naive	PS	PI
$\mathbb{E}\left[t_{\mathsf{exec}}\right]$ [s]	1.10^{-3}	1.10^{-2}	16.39	17.72	8.57	1.75

off-grid (PI/PS) approaches follow Algorithm 2, with the $\mathcal{G}_{\mathsf{L}}$ grid for the PS initialization being 16 times larger than that for the PI initialization, which explains its superior mean execution time. While the optimized method, i.e. off-grid (PI/PS), exhibits higher computational complexity compared to the classical fingerprinting approach, it offers a substantial improvement in localization accuracy. Moreover, as demonstrated in Table 5.1, the proposed off-grid approach reduces the number of required forward passes by a factor of twenty with respect to the on-grid approach, resulting in an almost tenfold reduction in mean execution time. This is further illustrated by comparing the computational complexity of the proposed method with the off-grid naive approach, which experiences a high mean execution time due to the absence of computational complexity optimization.

Memory requirements comparison

As shown in Fig. 5.7, the k-NN fingerprinting approach, that uses the neural generative model training dataset, achieves relatively satisfactory performance, with sub-meter median localization accuracy across all scenes. However, this approach necessitates the storage of $2\left|\mathcal{G}_{\mathsf{train}}\right| + 2N_a N_s \left|\mathcal{G}_{\mathsf{train}}\right|$ real scalars, i.e. training locations and associated complex channel coefficients, which, when evaluated on the S_2 training dataset, amounts to storing 121.3M real scalars. In comparison, the proposed localization method only requires storing the $|\boldsymbol{\theta}| = 9.1$M real learnable parameters of $f_{\boldsymbol{\theta}}$, resulting in a thirteenfold reduction in the required memory. This number of learnable parameters originates from the hyperparameters selected for the MB-$\tilde{\boldsymbol{\Psi}}_{\mathbf{a}}$ network in [Cha+25a]. Assuming a 32-bit floating-point representation, the fingerprinting approach demands 485.2 Mo of memory, whereas the proposed method only requires 36.4 Mo. This substantial memory reduction is achieved by eliminating the need for a large fingerprinting comparison dictionary. Importantly, this memory efficiency does not compromise localization accuracy, as the proposed method is not constrained by the number of stored coefficients, unlike classical fingerprinting methods, as highlighted in Eq. (5.6). The independence of the proposed method's localization performance from memory requirements is illustrated in Fig. 5.11. The proposed method consists of the off-grid (PI) approach, while the 3-NN results for different memory requirements are obtained by subsampling $\mathcal{D}_{\mathsf{tr}}$ on the S_2 scene.

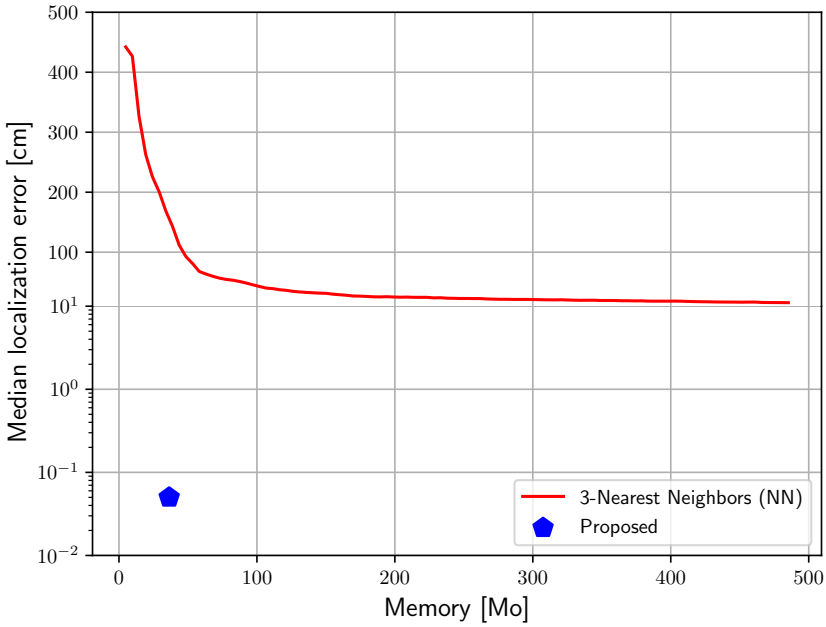


Figure 5.11: Memory-performance trade-off comparison. The memory requirements of the proposed method originates from the used neural network learnable parameters.

5.4 Conclusion

This chapter investigated the use of the model-based machine learning paradigm to perform user localization using radio signals, as formalized in Problem 3.2. The main takeaways can be summarized as follows:

- The neural architecture introduced in Chapter 4, designed to learn the location-to-channel mapping, was leveraged to enhance the localization accuracy of the well-known fingerprinting method, while simultaneously reducing its memory requirements

- Domain knowledge, formalized in (H1) was used to jointly **structure** and **optimize** a learning method. Indeed, the used neural architecture is structured according to a channel model derived from (H1), as detailed in Chapter 4. Furthermore, the improved localization procedure introduced in this chapter exploits the known distances between successive local minima of the considered similarity function, which are theoretically characterized using the same channel model.

- Simulations conducted on realistic data, encompassing both LoS and NLoS challenging radio environments, demonstrated that the proposed method achieves sub-wavelength localization accuracy, providing up to a two order of magnitude improvement over classical fingerprinting approaches.

- Unlike traditional fingerprinting methods, the performance of the proposed approach is not contingent on the size of a stored comparison dictionary. Instead, the neural model functions as a generative neural channel model, enabling on-the-fly inference of channel coefficients and thereby eliminating the need for an extensive pre-computed dictionary. This results in a tenfold reduction in memory requirements compared to classical fingerprinting techniques.

Chapter 6

Channel estimation and impairments learning

This chapter presents an optimized unfolded sparse recovery algorithm to perform channel estimation in SISO-OFDM systems, while learning hardware impairments. Section 6.1 introduces the problem and outlines the proposed contributions. Section 6.2 reviews the classical matching pursuit sparse recovery approach. Section 6.3 presents its unfolding into a neural network and details the architectural refinements aimed at reducing sample and time complexity. Section 6.4 reports experimental results, and Section 6.5 introduces concluding remarks.

Contents

6.1 Introduction **174**
- 6.1.1 General introduction 174
- 6.1.2 Contributions and related work 174
- 6.1.3 Problem formulation 175

6.2 Model-based approach **175**
- 6.2.1 Physical channel model 175
- 6.2.2 Sparse recovery approach 176
- 6.2.3 Hardware impairments 177

6.3 Proposed approach **178**
- 6.3.1 mpNet architecture 178
- 6.3.2 Reducing sample complexity: constrained dictionary 179
- 6.3.3 Reducing time complexity: hierarchical search 180

6.4 Experimental results **183**
- 6.4.1 Experimental setting 183
- 6.4.2 Results 185

6.5 Conclusion **188**

This chapter is mainly based on a conference article presented at the *2023 IEEE International Conference on Communications* [CLMR23]. Furthermore, the experiments described in this chapter entailed a total computational cost of approximately 2 GPU days.

6.1 Introduction

6.1.1 General introduction

Recently, machine learning techniques have emerged as a promising solution in many wireless communications areas [OH17; Wan+17] such as beamforming [LeMag+22b; Alk+18], channel charting [Stu+18; Fer+21; Yas+22] or channel mapping [LeMag21b; AA19]. The model-based machine learning paradigm, introduced in details in Chapter 3, and more specifically, the deep unfolding approach [Shl+23; MLE21; HRW14; BSS19] [1], offers a promising framework by combining the controlled complexity of classical signal processing approaches with the flexibility of machine learning techniques. Indeed, this approach considers that iterative algorithms can be unfolded as trainable neural networks, where each layer represents one iteration of the algorithm.

In the field of channel estimation, classical statistical estimation techniques such as the least squares (LS) or the minimum mean squared error (MMSE) have been extensively used in the past. However, these techniques suffer from several drawbacks: the LS estimator exhibits a significant mean squared error, whereas the MMSE estimator entails a substantial computational complexity. Alternative approaches have been proposed to address these drawbacks, one of which involves the use of sparse recovery algorithms. It is well known that propagation channels are dominated by a few strong propagation paths: the channel is then said to be sparse in a certain representation domain. It has been shown that this sparsity notion could be leveraged to design channel estimation strategies based on sparse recovery algorithms, such as Matching Pursuit (MP) [KI08; KY04; KGW09; YLM22]. However, these techniques rely on a priori knowledge of a physical wave propagation model, and it was shown in [YLM22] that a small uncertainty on system parameters could result in substantial estimation performance degradation.

6.1.2 Contributions and related work

Related work

Building on the aforementioned performance degradation issue, the authors in [YLM22] proposed leveraging the deep unfolding approach through the unfolded network mpNet, to perform channel denoising while accounting to antenna hardware impairments. This approach demonstrated improved denoising performance compared to classical statistical estimation techniques, while simultaneously compensating HWIs. However, this method introduces a large number of learnable parameters, resulting in high sample and time complexities.

1. See Section 3.2.1 for a detailed introduction.

Contributions

It is proposed to tackle the sample and time complexity issues of the mpNet neural network by refining its architecture. Additionally, while the study in [YLM22] focused on a multi-antenna system with a single carrier, this chapter focuses on a multi-carrier system with a single antenna. Nevertheless, the proposed contributions are applicable to both system configurations. The main contributions of this chapter are summarized as follows:

- The sample complexity of mpNet is reduced by introducing a constrained dictionary approach. Specifically, rather than learning an entire matrix that is affected by hardware impairments, it is proposed to only learn the HWI-dependent parameters and subsequently compute the matrix corresponding to the learned parameter values.
- The time complexity of mpNet is optimized by introducing a hierarchical search procedure. In particular, instead of performing correlations with all columns of a large matrix, it proposed to iteratively refine the search space in a divide and conquer manner, to identify the most relevant column.

6.1.3 Problem formulation

This chapter focuses on a SISO-OFDM scenario with N subcarriers in the uplink. The different subcarrier frequencies are denoted as $\mathbf{f} \in \mathbb{R}^N$, and the BS antenna gain at each subcarrier frequency are represented by $\mathbf{g} \in \mathbb{C}^N$. Finally, a propagation channel with L_p propagation paths is considered between the different UEs and BS.

Let $\mathbf{x} \in \mathbb{C}^N$ be an LS channel estimate of the frequency domain channel vector $\mathbf{h} \in \mathbb{C}^N$. Due to the imperfect nature of the LS estimator, $\mathbf{x}$ is affected by a residual estimation error:

$$\mathbf{x} = \mathbf{h} + \mathbf{n}, \tag{6.1}$$

where $\mathbf{n} \sim \mathcal{CN}\left(\mathbf{0}_{\mathbb{R}^N}, \sigma^2 \mathbf{Id}_N\right)$. Its SNR can be computed as:

$$\mathsf{SNR}_{\text{in}} = \frac{\|\mathbf{h}\|_2^2}{N\sigma^2}. \tag{6.2}$$

The goal of this chapter is to denoise this LS channel estimate using the model-based machine learning paradigm.

6.2 Model-based approach

6.2.1 Physical channel model

For a given subcarrier f_k, the uplink channel coefficient between a given UE and BS can be expressed as:

$$h_k = \sum_{l=1}^{L_p} \alpha_l g_k \mathrm{e}^{-\mathrm{j}2\pi f_k \tau_l}, \tag{6.3}$$

where α_l and τ_l respectively represent the complex attenuation coefficient and the propagation delay for the lth path. Under this model, the channel vector $\mathbf{h} \in \mathbb{C}^N$ can be expressed as:

$$\mathbf{h} = \sum_{l=1}^{L_p} \alpha_l \begin{bmatrix} g_1 \mathrm{e}^{-\mathrm{j}2\pi f_1 \tau_l} \\ \vdots \\ g_N \mathrm{e}^{-\mathrm{j}2\pi f_N \tau_l} \end{bmatrix} = \sum_{l=1}^{L_p} \alpha_l \boldsymbol{\psi}(\tau_l), \tag{6.4}$$

where $\boldsymbol{\psi}(\tau_l)$ is the frequency response vector of the lth path. As shown in Eq. (6.4), the channel vector can be defined as a linear combination of FRVs. This combination is considered sparse when the number of paths L_p is small.

Remark 6.1. *The propagation channel model presented in Eq.* (6.4) *directly originates from the domain knowledge formalized in (H1). Indeed, it can be seen as a simplified version of the model presented in Theorem 4.2, which was explicitely derived from (H1).*

6.2.2 Sparse recovery approach

As presented above, the propagation channel can be represented as a sparse linear combination of FRVs. As such, it is possible to construct a fixed matrix $\boldsymbol{\Psi} \in \mathbb{C}^{N \times A}$ with A different FRV columns, which amounts to the discretization of the delay space $\mathbb{R}^+$, and a coefficients vector $\mathbf{u} \in \mathbb{C}^A$ such that the channel could be estimated as:

$$\hat{\mathbf{h}} = \boldsymbol{\Psi}\mathbf{u}. \tag{6.5}$$

A sparse recovery problem consists in identifying an activation vector $\mathbf{u}$, that activates only a limited number of FRVs, i.e. that is sparse. As is common in the sparse recovery literature, the FRV matrix $\boldsymbol{\Psi}$ will hereafter be referred to as the *dictionary*, and its columns as *atoms*, i.e. each atom represents a FRV associated with a specific delay. This sparse recovery problem can be formalized as follows for a fixed dictionary $\boldsymbol{\Psi}$:

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & \|\mathbf{h} - \boldsymbol{\Psi}\mathbf{u}\|_2^2 . \\ \text{subject to} \quad & \|\mathbf{u}\|_0 \leq A \end{aligned} \tag{6.6}$$

This optimization problem makes sense only if the dictionary is correct, i.e. a sparse linear combination of dictionary columns can represent a channel vector. If this is not the case, the channel estimate won't be trustworthy. It is possible to find the optimal dictionary as:

$$\boldsymbol{\Psi}^\star = \arg\min_{\boldsymbol{\Psi}} \mathbb{E}_{\mathbf{h}} \left[\min_{\mathbf{u}, \|\mathbf{u}\|_0 \leq A} \|\mathbf{h} - \boldsymbol{\Psi}\mathbf{u}\|_2^2 \right]. \tag{6.7}$$

Solving Eq. (6.6) for a fixed FRV dictionary is known to be computationally challenging [2]. Consequently, several surrogate approaches have been proposed, including convex relaxation techniques and greedy algorithms. This chapter, following the methodology

2. It has been shown to be NP-hard: an interested reader is referred to [Nat95; TW10].

in [YLM22], focuses on the latter, with a particular emphasis on the MP algorithm [MZ93]. The goal of this algorithm is to iteratively project the noisy channel onto the most correlated atoms, so as to denoise it. After L_p iterations, under ideal conditions, all corresponding FRV contributions to the channel will have been subtracted from the noisy channel estimate. Consequently, the residual will consist solely of noise, allowing to easily perform denoising. The MP algorithm is presented in details in Algorithm 3.

Algorithm 3 MP algorithm for channel denoising.

Inputs: Dictionary $\mathbf{\Psi}$, Noisy LS estimate $\mathbf{x}$, Tolerance level ϵ

1: Initialize the residual: $\mathbf{r} \leftarrow \mathbf{x}$
2: **while** $\|\mathbf{r}\|_2^2 > \epsilon$ **do**
3: Find the most correlated atom: $i^\star \leftarrow \arg\max_i \left|\boldsymbol{\psi}_i^{\mathsf{H}}\mathbf{r}\right|$
4: Update the residual: $\mathbf{r} \leftarrow \mathbf{r} - \boldsymbol{\psi}_{i^\star}\boldsymbol{\psi}_{i^\star}^{\mathsf{H}}\mathbf{r}$
5: **end while**

Output: $\hat{\mathbf{h}} \leftarrow \mathbf{x} - \mathbf{r}$ (Denoised LS estimate)

6.2.3 Hardware impairments

In realistic conditions, the FRVs are not exactly known as there exists a lot of hardware impairments related to frequency generation/acquisition. More specifically, sampling clock offset (SCO) can offset the subcarrier frequencies. Let $\tilde{f}_i$ be the *nominal* ith subcarrier frequency, i.e. the non-offsetted subcarrier frequency, and f_i be the *physical* ith subcarrier frequency, i.e. the potentially offsetted subcarrier frequency. For the SCO, it has been shown in [Kim+98; Spe+99] that the frequency offset is dependent on the subcarrier index, on the oscillator part per million (ppm) value ξ, and on the subcarrier spacing Δf:

$$\forall i \in \left[\!\left[-\frac{N}{2}, \frac{N}{2}\right]\!\right], \; f_i = \tilde{f}_i + i\delta f = \tilde{f}_i + i\xi\Delta f. \tag{6.8}$$

Moreover, an antenna gain imperfection is considered. Let $\tilde{g}_i$ be the nominal antenna gain, g_i, the physical antenna gain for the ith subcarrier, can be expressed as:

$$\forall i \in \left[\!\left[-\frac{N}{2}, \frac{N}{2}\right]\!\right], \; g_i = \tilde{g}_i + n_{g_i}, \tag{6.9}$$

with $n_{g_i} \sim \mathcal{N}\left(0, \sigma_g^2\right)$.

The notions of *nominal* and *physical dictionaries* can now be defined. The nominal dictionary $\tilde{\mathbf{\Psi}}$ represents the dictionary constructed from the nominal knowledge of the system parameters, which is unaware of impairments. On the other hand, the physical dictionary represents the dictionary constructed from the perfect knowledge of the system parameters, i.e. with full knowledge of the SCO and antenna gain imperfections. The

proposed method consists in initializing a neural network using the nominal dictionary, and approaching the performance one would get using the physical dictionary via learning.

6.3 Proposed approach

6.3.1 mpNet architecture

Using the same approach as in [YLM22], it is proposed to use the unfolded neural network mpNet to achieve SISO-OFDM channel estimation, while learning the HWIs. This network consists in the unfolded version of the MP iterative algorithm, and its forward pass is presented in Algorithm 4.

Algorithm 4 Forward pass of mpNet [YLM22].

Inputs: Weight matrix $\mathbf{W} \in \mathbb{C}^{N \times A}$, Noisy LS estimate $\mathbf{x} \in \mathbb{C}^{N}$, Noise variance σ^2, Number of subcarriers N

1: $\mathbf{r} \leftarrow \mathbf{x}$
2: $\epsilon \leftarrow \frac{\sigma^2 N}{\|\mathbf{x}\|_2^2}$
3: **while** $\|\mathbf{r}\|_2^2 > \epsilon$ **do**
4: $\quad \mathbf{r} \leftarrow \mathbf{r} - \mathbf{W}\left(\mathrm{HT}_1\left(\mathbf{W}^{\mathsf{H}}\mathbf{r}\right)\right)$
5: **end while**

Output: $\hat{\mathbf{h}} \leftarrow \mathbf{x} - \mathbf{r}$ (Denoised LS estimate)

The links between MP and its unfolded version mpNet clearly appear in the equation $\mathbf{W}\left(\mathrm{HT}_1\left(\mathbf{W}^{\mathsf{H}}\mathbf{r}\right)\right)$. First, the correlation term $\mathbf{W}^{\mathsf{H}}\mathbf{r}$ corresponds to the dictionary correlation step. Second, the operator HT_1 denotes a non-linear hard-thresholding operator which keeps only its input of greatest modulus and sets all the others to zero: it is used to select the most correlated atom. Specifically, one obtains:

$$\mathrm{HT}_1\left(\mathbf{W}^{\mathsf{H}}\mathbf{r}\right) = \begin{bmatrix} 0 & \cdots & 0 & \mathbf{w}_{i^\star}^{\mathsf{H}}\mathbf{r} & 0 & \cdots & 0 \end{bmatrix}^{\mathsf{T}}, \tag{6.10}$$

where $i^\star$ denotes the index of the most correlated atom. Third, the matrix product with $\mathbf{W}$ allows to perform the projection step. Indeed:

$$\begin{aligned} \mathbf{W}\mathrm{HT}_1\left(\mathbf{W}^{\mathsf{H}}\mathbf{r}\right) &= \mathbf{W}\begin{bmatrix} 0 & \cdots & 0 & \mathbf{w}_{i^\star}^{\mathsf{H}}\mathbf{r} & 0 & \cdots & 0 \end{bmatrix}^{\mathsf{T}} & (6.11)\\ &= \mathbf{w}_{i^\star}\left(\mathbf{w}_{i^\star}^{\mathsf{H}}\mathbf{r}\right) & (6.12)\\ &= \mathbf{P}_{\mathbf{w}_{i^\star}}\mathbf{r}, & (6.13) \end{aligned}$$

where $\mathbf{P}_{\mathbf{w}_{i^\star}} = \mathbf{w}_{i^\star}\mathbf{w}_{i^\star}^{\mathsf{H}}$ denotes a projection matrix, since atoms are of unit-norm.

Owing to its differentiable nature, mpNet can be used to learn a dictionary that accounts for the impact of HWIs through gradient descent. The weight matrix $\mathbf{W}$ is

therefore initialized with the nominal dictionary $\tilde{\mathbf{\Psi}}$. This thoughtful initialization reduces the convergence time of the loss function compared to a random initialization, and can be interpreted as the incorporation of a priori domain knowledge in the neural network initialization. To summarize, this network pursues two main objectives:

- The forward pass of mpNet performs the channel estimation/denoising operation as the architecture of mpNet is the unfolded MP algorithm. Each layer will estimate one path of the channel.
- The backward pass of mpNet performs the dictionary learning. As the weight matrix of mpNet is initialized with the nominal dictionary, the goal of the backward pass is to update the weight matrix so that it tends towards the physical dictionary.

In Algorithm 4 the value ϵ corresponds to a stopping criterion denoted as $\mathtt{SC}_2$ in [YLM22]:

$$\mathtt{SC}_2 : \|\mathbf{r}\|_2^2 \leq \frac{\sigma^2}{\|\mathbf{x}\|_2^2} N. \tag{6.14}$$

This stopping criterion allows the network to have an adaptive number of layers for given noise characteristics: in other words, SNR adaptability. However, it imposes the knowledge of the noise variance σ^2 at the BS side.

6.3.2 Reducing sample complexity: constrained dictionary

As previously explained, the learnable parameters of mpNet are the components of the weight matrix $\mathbf{W} \in \mathbb{C}^{N \times A}$. Owing to the complex nature of $\mathbf{W}$, there are $2NA$ learnable parameters. In the experimental setup that will be presented in Section 6.4, $N = 256$ subcarriers and $A = 990$ atoms are considered, which results in $2NA = 506,880$ learnable parameters. The first contribution of this chapter consists in reducing this number by using a *constrained* dictionary approach.

The studied SISO-OFDM system considers only imperfections affecting the gains and frequencies of the dictionary. Moreover, since these gains and frequencies are expected to be independent of the propagation delays, they are shared across all atoms of the dictionary. Based on this observation, a constrained dictionary can be defined as follows:

$$\overset{\diamond}{\mathbf{\Psi}} = \begin{bmatrix} g_1 e^{-\mathrm{j}2\pi\left(f_1 - \frac{N}{2}\delta f\right)\tau_1} & \dots & g_1 e^{-\mathrm{j}2\pi\left(f_1 - \frac{N}{2}\delta f\right)\tau_A} \\ \vdots & \vdots & \vdots \\ g_N e^{-\mathrm{j}2\pi\left(f_N + \frac{N}{2}\delta f\right)\tau_1} & \dots & g_N e^{-\mathrm{j}2\pi\left(f_N + \frac{N}{2}\delta f\right)\tau_A} \end{bmatrix} \in \mathbb{C}^{N \times A}. \tag{6.15}$$

Each column of the dictionary still represents an FRV, however the components of this constrained matrix are no more learnable. The only learnable parameters are the complex antenna gains $\mathbf{g} \in \mathbb{C}^N$ and the SCO frequency offset $\delta f \in \mathbb{R}$. Consequently, the number of learnable parameters is reduced from $2NA$ to $2N + 1$, making it independent of the

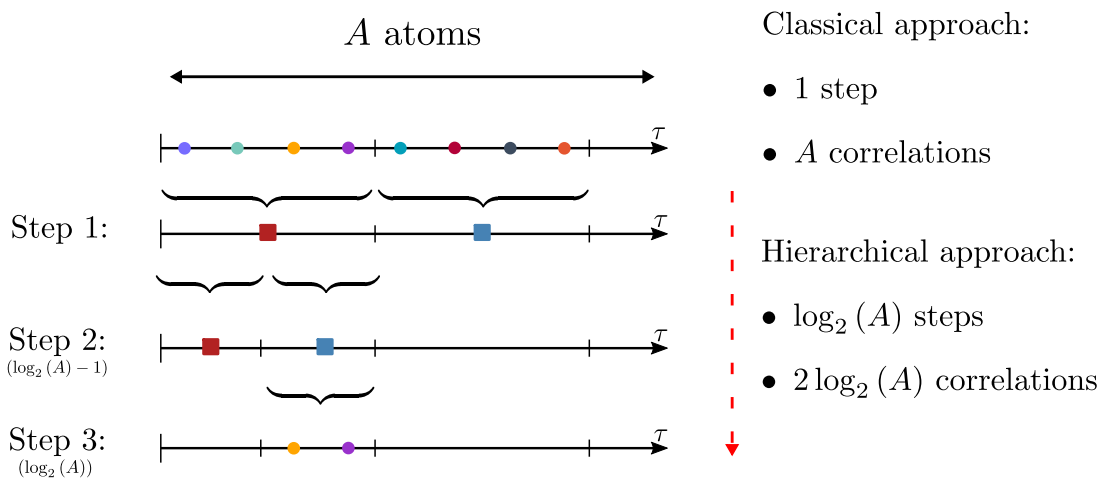


Figure 6.1: Hierarchical atom search.

number of atoms. In the numerical example considered earlier, this corresponds to a reduction from $2NA = 506,880$ parameters to only $2N + 1 = 513$.

Another advantage of the proposed constrained dictionary approach lies in its sample complexity, i.e. the number of training samples required to achieve convergence of the network's loss function. In the classical mpNet architecture, since each element of the weight matrix is independent, only the atoms selected during the forward pass are updated through backpropagation. In contrast, in the proposed approach, the frequency offset and antenna gains are shared across all atoms. Consequently, updating a single physical parameter, i.e. antenna gain or frequency offset, results in an update of all atoms, leading to faster convergence of the loss function, compared to the classical mpNet architecture, as demonstrated in Section 6.4.

6.3.3 Reducing time complexity: hierarchical search

A quick complexity analysis shows that the time complexity of the classical mpNet approach is $\mathcal{O}(DNA)$ where D represents the number of layers of the mpNet network [3]. In the context of the previously considered scenario with $D = 10$ layers, $DNA = 2,534,400$ arithmetic operations are required for the estimation of one SISO-OFDM channel. To mitigate this computational burden, a new hierarchical atom search is proposed.

Hierarchical atom search

The principle of the hierarchical atom search approach is presented in Fig. 6.1. The classical approach consists in finding the most correlated atom in a dictionary of A different atoms (represented as circles in Fig. 6.1): as such, A different correlations are required. The

3. Here, DNA denotes the total number of multiplications required to compute the correlations for A atoms of size N across D layers.

proposed hierarchical approach consists in iteratively finding the most correlated atom in the dictionary, with less required correlations. As in the SISO-OFDM scenario the atoms are parameterized with a propagation delay, the dictionary covers a specific interval in the delay domain. In the first step, two *meta-atoms* are constructed (represented as squares in Fig. 6.1), such that their correlation responses separately cover the dictionary-associated delay interval. The correlations with both meta-atoms are computed, and the one yielding the highest response is selected. Subsequently, two new meta-atoms are generated to cover the delay interval corresponding to the previously selected meta-atom, and their correlations are computed. This hierarchical process is iteratively repeated until step $\log_2(A)$, where the delay interval associated with the most correlated meta-atom from step $\log_2(A) - 1$ contains only two atoms from the original dictionary. Their correlations are computed, and the atom with the highest correlation is identified. This approach enables the selection of the most correlated atom in $\log_2(A)$ steps, but requires only $2\log_2(A)$ correlations.

Meta-atom design

The main challenge of the proposed hierarchical approach lies in designing a meta-atom whose correlation response effectively spans a specific delay interval. It can be shown that using a cardinal sine modulated FRV as a meta-atom satisfies this requirement. First, a Fourier transform from the frequency domain to the delay domain is defined as:

$$x(\tau) = \mathcal{F}_f[x(f)] = \int_{\mathbb{R}} x(f)\,\mathrm{e}^{-\mathrm{j}2\pi f\tau}\mathrm{d}f. \tag{6.16}$$

Denoting the bandwidth, central frequency, and subcarrier spacing of the system as γ, f_0, and Δf, the correlation of a pure FRV of delay τ, $e(f,\tau)$, and a generic meta-atom $\psi_i(f)$ can be computed as:

$$\chi_{\psi_i}(\tau) = \langle e(f,\tau), \psi_i(f)\rangle \tag{6.17}$$

$$= \int_{\mathbb{R}} \psi_i^*(f)\, g(f)\,\mathrm{e}^{-\mathrm{j}2\pi f\tau}\Pi_\gamma(f-f_0)\,Ш_{\Delta f}(f)\,\mathrm{d}f. \tag{6.18}$$

The windowing $\Pi_\gamma(f-f_0)$ allows to represent the band-limited nature of the FRV and meta-atom, while the Dirac comb $Ш_{\Delta f}(f)$ allows the sampling at the subcarrier frequencies. The antenna gains are assumed to be quasi-constant over the bandwidth so that $g(f) = \alpha\Pi_\gamma(f-f_0)$. Note that Section 6.4 presents an assessment of the robustness of the proposed method on channels that present non-constant gains over the bandwidth. In order to simplify the following developments, the constant α is deliberately omitted. Observing that Eq. (6.18) represents a Fourier transform, one then obtains:

$$\chi_{\psi_i}(\tau) = \mathcal{F}_f[\psi_i^*(f)\,\Pi_\gamma(f-f_0)\,Ш_{\Delta f}(f)]. \tag{6.19}$$

Since $\chi_{\psi_i}(\tau)$ is desired to be constant over a specific delay interval (i.e. a rectangular function), it is straightforward to see that modulating FRVs with a cardinal sine function satisfies this requirement. More specifically, the meta-atom $\psi_i(f)$ can be defined as:

$$\psi_i(f) = \operatorname{sinc}(\varpi f)\, \mathrm{e}^{-\mathrm{j}2\pi f \tau_i}. \tag{6.20}$$

The correlation response then becomes:

$$\chi_{\psi_i}(\tau) = \mathcal{F}_f\left[\operatorname{sinc}(\varpi f)\, \mathrm{e}^{\mathrm{j}2\pi f \tau_i} \Pi_\gamma(f - f_0)\, \text{Ш}_{\Delta f}(f)\right] \tag{6.21}$$

$$= \tfrac{1}{\varpi}\Pi_\varpi(\tau - \tau_i) \circledast \gamma \operatorname{sinc}(\gamma\tau)\, \mathrm{e}^{-\mathrm{j}2\pi f_0 \tau} \circledast \tfrac{1}{\Delta f}\text{Ш}_{\frac{1}{\Delta f}}(\tau). \tag{6.22}$$

One can see in Eq. (6.22) that the proposed meta-atom expression allows to obtain a rectangular waveform in the delay domain, ϖ controlling the width, and τ_i the central position. This is illustrated in Fig. 6.2 where the correlation responses are presented for different values of ϖ and τ_i [4].

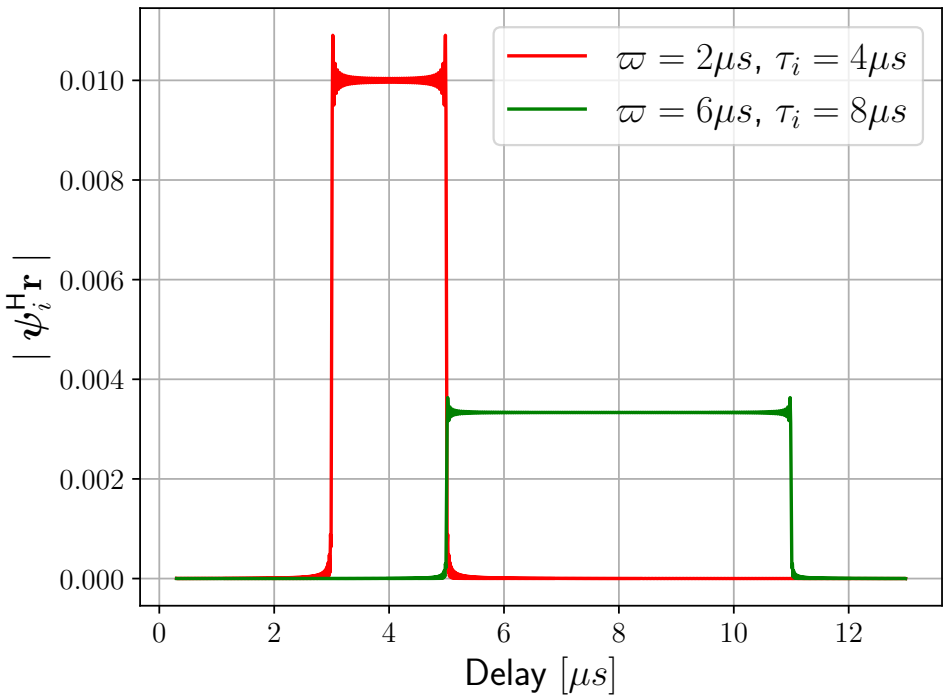


Figure 6.2: Influence of ϖ and τ_i on the correlation waveform.

4. The oscillations observed at the edges of the rectangular waveforms are a manifestation of the well-known Gibbs phenomenon.

Table 6.1: Unfolded network complexity analysis.

		mpNet	C. mpNet	H.C.3 mpNet
Time complexity	Forward	$\mathcal{O}(DNA)$	$\mathcal{O}(DNA)$	$\mathcal{O}(DN3\log_3(A))$
	Backward	$\mathcal{O}(DN)$	$\mathcal{O}(DN)$	$\mathcal{O}(DN)$
Sample complexity		+++	+	+

Time complexity

For two meta-atoms, the time complexity of the proposed hierarchical search can be evaluated as $\mathcal{O}(2\log_2(A))$. Considering the partition of $A > 1$ atoms into $n \in \mathbb{N}$, $n > 1$ intervals, the time complexity of the associated hierarchical search is $\mathcal{O}(n\log_n(A))$. Minimizing the time complexity of this hierarchical approach is then equivalent to solving Eq. (6.23):

$$n^\star = \underset{n\in\mathbb{N},\, n>1}{\arg\min}\, n\log_n(A)\,. \tag{6.23}$$

It can be easily shown that solving Eq. (6.23) gives $n^\star = 3$.

A quick complexity analysis of all the unfolded networks presented so far is presented in Table 6.1. It is proposed to compare the classical mpNet as it was introduced in [YLM22], mpNet with a constrained dictionary (C. mpNet), and mpNet with a constrained dictionary and hierarchical atom search with 3 meta-atoms (H.C.3 mpNet). As stated earlier, the constrained dictionary approach reduces the sample complexity, while the hierarchical search approach was introduced to reduce the time complexity. Combining both methods is therefore particularly appealing, as it enables a joint reduction of both sample and time complexities. For $D = 10$ layers, $N = 256$ subcarriers and $A = 990$ atoms, the time complexity of the forward pass goes from $2,534,400$ arithmetic operations for the classical mpNet to around only $48,220$ arithmetic operations for H.C.3 mpNet, which is a fiftyfold reduction.

6.4 Experimental results

6.4.1 Experimental setting

Data generation and simulation parameters

It is proposed to validate the contributions introduced in this chapter using realistic channel data, namely the "O1" outdoor `DeepMIMO` dataset [Alk19]. More specifically, the used BS is represented with a red triangle in Fig. 6.3, and UEs are randomly generated in the blue zone. A central frequency of 3.4GHz, and a bandwidth of 50MHz are considered. Furthermore, the channels are generated with $N = 256$ subcarriers. The dictionary is

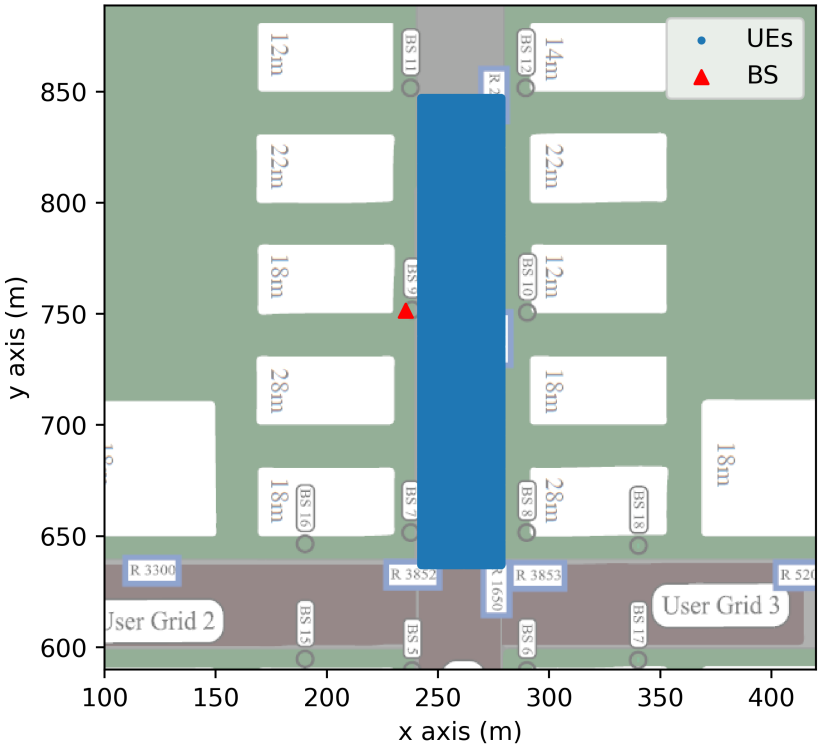


Figure 6.3: `DeepMIMO` channel generation configuration.

composed of $A = 990$ atoms. Note that up to $L_p = 10$ propagation paths are considered. Regarding the physical imperfections, the oscillator ppm value for the SCO is $\xi = 40$ppm, and two gain noise variance scenarios are considered: $\sigma_g^2 = 0.36$ (very high gain noise) and $\sigma_g^2 = 0.09$ (high gain noise). Two SNR scenarios are distinguished: $\mathsf{SNR}_{\text{in}} = 10$dB and $\mathsf{SNR}_{\text{in}} = 5$dB, knowing that the conclusions hold for lower SNRs. In order to compute the stopping criterion for the network, the noise variance is assumed to be known at the BS side.

Learning strategy

The same online unsupervised learning strategy as in [YLM22] is used to train the proposed neural architecture. The evaluation metric is the NMSE, defined as:

$$\mathsf{NMSE} = \frac{1}{|\mathcal{T}|} \sum_{\mathbf{h} \in \mathcal{T}} \frac{\left\| \hat{\mathbf{h}} - \mathbf{h} \right\|_2^2}{\|\mathbf{h}\|_2^2}, \tag{6.24}$$

where $\mathcal{T}$ denotes the test channel set, $\mathbf{h}$ denotes the ground-truth channels, and $\hat{\mathbf{h}}$ the estimated channels.

6.4.2 Results

Performance comparison

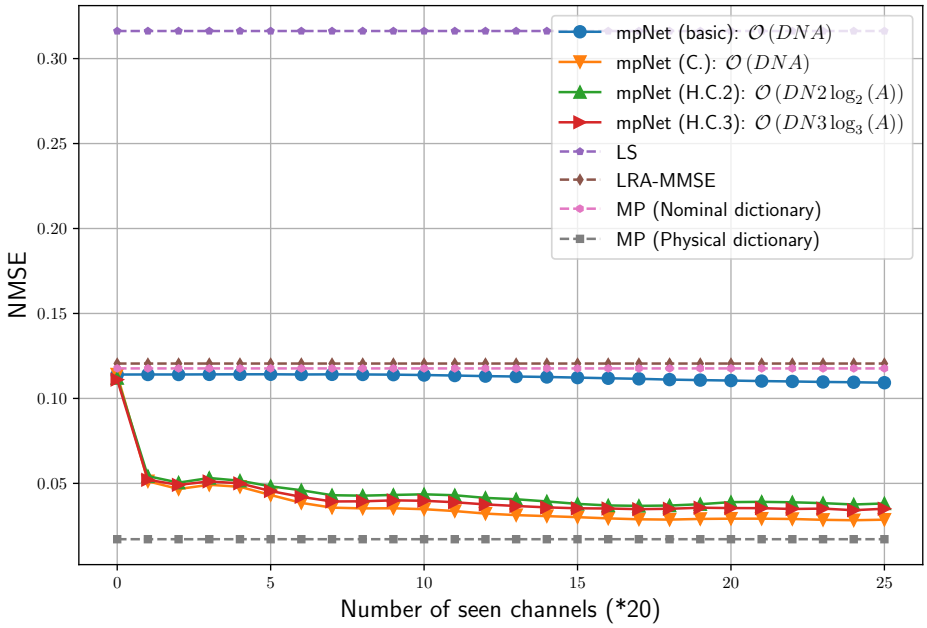
A comparative performance analysis is conducted across four configurations: the classical network (mpNet), the constrained dictionary mpNet (C. mpNet), the hierarchical search with 2 meta-atoms and constrained dictionary mpNet (H.C.2 mpNet) and the one with 3 meta-atoms (H.C.3 mpNet). Additionally, these network performance are compared against the LS estimator, the MP algorithm with nominal and physical dictionaries, and the MMSE estimator with a substituted channel covariance matrix using the low-rank approximation (LRA) presented in [SL17]. The substituted covariance matrix is used because, in practice, estimating the covariance matrix for each UE is too complex, as it would require tracking each UE's channel over multiple frames. Several comments can be made on the results presented in Fig. 6.4, as listed below.

- **Dictionary learning.** For all noise and hardware scenarios, dictionary learning is achieved as there is convergence of the NMSE performance. One can remark that at the end of the training, the proposed networks (C. mpNet, H.C.2 and H.C.3 mpNet) always outperform the LS and LRA-MMSE approaches.

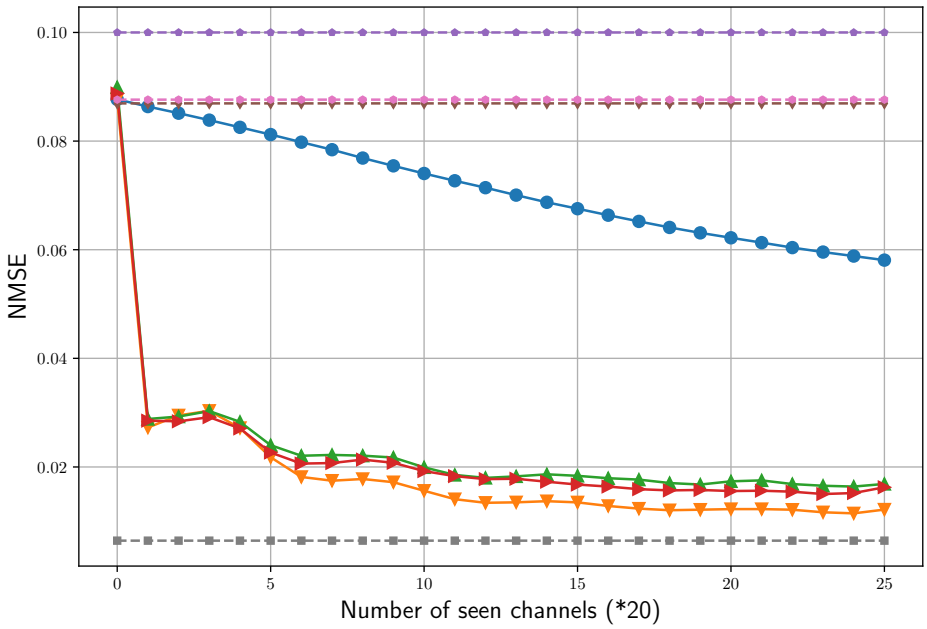
- **Convergence speed.** As exposed in Section 6.3.2, the convergence of the constrained versions of mpNet (C. mpNet and H.C. mpNet) is quicker than the convergence of the classical mpNet. For all scenarios, only approximately 400 channels are needed to achieve convergence. This can be explained by the reduced number of learnable parameters and their sharing across the atoms of the dictionary in the constrained versions of mpNet.

- **Estimation performance** After convergence, the estimation performance of the constrained networks are better than the one of the classical mpNet. This can be explained by the bias-variance decomposition of the estimation NMSE. In fact, the C./H.C. mpNet can be viewed as an estimator with fewer parameters than the classical mpNet, which results in a lower variance. As the dictionary structure is wisely chosen, the C./H.C. mpNet exhibits the same bias as the classical mpNet. Consequently, both the C. and H.C. mpNet architectures achieve a lower estimation NMSE compared to the classical one. However, the proposed unfolded approach allows to compensate for these imperfections, and present satisfactory performance.

- **Large HWIs.** One can remark in Fig. 6.4 (c) that, if the hardware imperfections are too important, the model-based MP algorithm does not perform well. In that particular case, using the MP approach with nominal dictionary is worse than using LS. However, it can be observed that the proposed method can handle such large impairments, as it converges towards the performance of MP with the physical dictionary, and clearly outperforms the LS method.

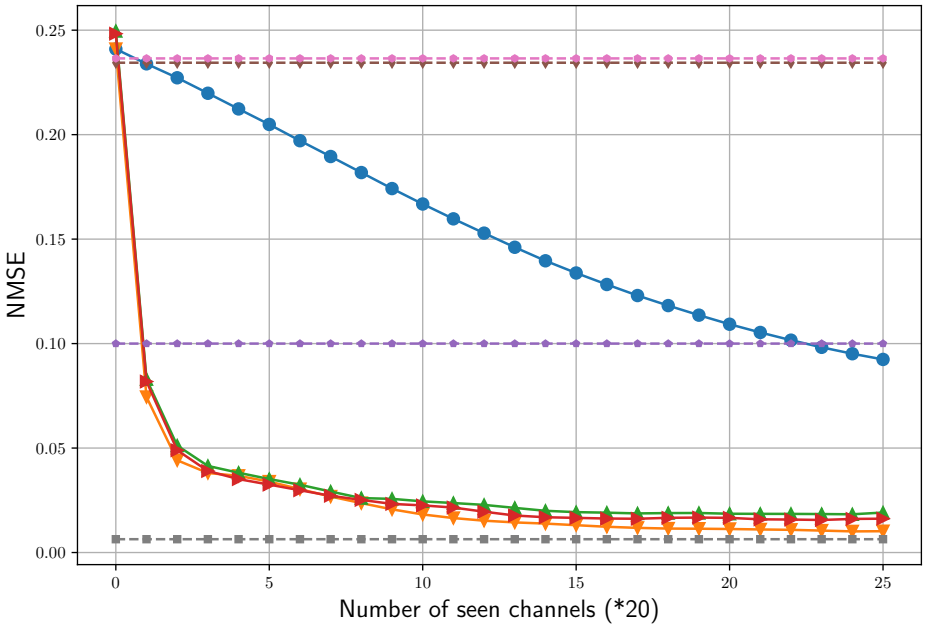


(a) $\mathsf{SNR}_{\text{in}} = 5\,\text{dB}$, $\sigma_g^2 = 0.09$, $\xi = 40\,\text{ppm}$.



(b) $\mathsf{SNR}_{\text{in}} = 10\,\text{dB}$, $\sigma_g^2 = 0.09$, $\xi = 40\,\text{ppm}$

Figure 6.4: Channel estimation performance for various SNRs and model imperfections.



(c) $\mathsf{SNR}_{\text{in}} = 10\,\text{dB}$, $\sigma_g^2 = 0.36$, $\xi = 40\,\text{ppm}$.

Figure 6.4: Channel estimation performance for various SNRs and model imperfections (cont.).

- **H.C./C. mpNet comparison.** It is noticeable that H.C.2 mpNet offers slightly worse performance than C. mpNet. This behavior can be explained by the hierarchical search approach. In the classical method, the algorithm directly identifies the most correlated atom of the dictionary. In contrast, the hierarchical approach first selects the delay interval containing the highest energy. As a result, for a fixed noisy LS estimate, the two approaches may not select the same atoms, leading to different channel estimation performance. It is then interesting to study the impact of the number of meta-atoms, as increasing the number of intervals in the delay domain can improve the estimation performance for the hierarchical approach. This can be observed with the NMSE performance of H.C.3 mpNet, which achieves slightly better performance than H.C.2 mpNet while maintaining a lower time complexity, as presented in Table 6.1.

Execution time

It has been shown in Section 6.3.3 [5] that the hierarchical approach with 3 meta-atoms was optimal in terms of time complexity. This can be seen in Table 6.2 which presents an

5. See Eq. (6.23) for more information.

Table 6.2: Average execution time for 10^4 random UEs, and varying number of atoms.

		MP	MP H.2	MP H.3
$\mathbb{E}\left[t_{\text{exec}}\right]$ [s]	$A = 990$	0.048	0.014	**0.013**
	$A = 5,000$	0.263	0.019	**0.015**
	$A = 50,000$	2.866	0.024	**0.020**

execution time comparison between the MP algorithm and its hierarchical versions with 2 and 3 meta-atoms, namely MP H.2 and MP H.3. One can see that MP H.3 outperforms both the MP H.2 and classical MP algorithm, with an average execution time up to a hundred time quicker than the classical MP.

6.5 Conclusion

This chapter focused on the use of the MB-ML paradigm for joint channel estimation and hardware impairments learning, as formalized in Problem 3.3. Structural modifications to the neural network mpNet, introduced in [YLM22], were proposed to optimize both its sample and time complexity. The main takeaways can be summarized as follows.

- The proposed channel denoising method exploits the sparsity of a propagation channel model, grounded in the domain knowledge formalized in (H1). Accordingly, the MB-ML paradigm, and more specifically, deep-unfolding, was leveraged to **structure** a neural architecture. Additionally, domain knowledge was also used to **initialize** the structure, through the use of nominal frequency response vectors in the dictionary used in the unfolded MP network.

- The sample complexity of mpNet was reduced through a constrained dictionary approach. Instead of learning a full dictionary of frequency response vectors, only the parameters associated with hardware impairments that affect this dictionary were learned, leading to nearly a thousandfold reduction in the number of learnable parameters.

- The time complexity of mpNet was optimized via a hierarchical search approach. Instead of computing correlations with every column, i.e. atom, of the dictionary to find the most correlated one, meta-atoms were introduced to iteratively select two candidate atoms, with less required correlations. This resulted in a hundredfold reduction in the average inference time.

- The estimation performance of the proposed method was benchmarked against several baselines. Results show that the enhanced mpNet network outperforms both the original mpNet and classical baselines, in terms of estimation performance as well as convergence speed.

Chapter 7

DoA estimation and impairments learning

This chapter presents a differentiable version of the MUSIC algorithm, enabling joint hardware impairments learning and direction-of-arrival estimation. Section 7.1 introduces the problem setting and discusses the impact of hardware impairments on the estimation process. Section 7.2 provides a formal presentation of the MUSIC algorithm along with its theoretical foundations. Section 7.3 then highlights the inherent non-differentiability of MUSIC and introduces a modified differentiable formulation. Finally, Section 7.4 demonstrates the superior performance of the proposed method compared to multiple baselines, and Section 7.5 offers concluding remarks.

Contents

7.1 Problem setting **190**
- 7.1.1 General introduction 190
- 7.1.2 Related work and contributions 190
- 7.1.3 Problem formulation 191

7.2 Model-based approach **192**
- 7.2.1 Subspace methods 192
- 7.2.2 MUSIC 193

7.3 Towards diffMUSIC **194**
- 7.3.1 Model-based differentiable architecture 195
- 7.3.2 Task-based loss functions 197

7.4 Experiments **198**
- 7.4.1 Learning framework and baselines 199
- 7.4.2 Results 199

7.5 Conclusion **204**

This chapter is mainly based on a conference article presented at the *2025 IEEE International Conference on Communications* [Cha+25c]. Furthermore, the experiments described in this chapter entailed a total computational cost of approximately 25 GPU days.

7.1 Problem setting

7.1.1 General introduction

Direction of arrival estimation refers to the angular estimation of a wavefront impinging on a sensor array. In modern wireless communication systems, DoAs are used for a wide variety of tasks: localization, target tracking, beamforming, or interference management [PTHV23; CNG24]. The growing interest in integrated sensing and communication (ISAC), where sensing and communication are jointly performed, has further reinforced the need of efficient DoA estimation methods. Classically, model-based algorithms have been proposed to solve the DoA estimation problem [Sch86; Bar83; VVB88; RK89; Vas09]. While the use of subspace methods such as multiple signal classification (MUSIC) [Sch86], or estimation of signal parameters via rotational invariance techniques (ESPRIT) [RK89] allows to increase estimation performance, such methods are known to be sensitive to hardware impairments. Several studies have been carried out to assess the impact of hardware impairments on wireless communication performance [Bjö+13; Bjö+14; Bjö+15; HR10; CC22a], but also on sensing performance [Boz+21; TI24; Che+24].

Owing to its inherent interpretability and efficiency advantages, as well as the broad availability of analytical models, the MB-ML paradigm has been applied to a wide range of tasks in the wireless communications field. This chapter explores whether MB-ML can be effectively applied to jointly learn hardware impairments and estimate DoAs from measurements, while maintaining both efficiency and interpretability, i.e. the task described in Problem 3.3.

7.1.2 Related work and contributions

Related work

The use of machine learning methods in the DoA estimation context has been studied in the literature: a non-exhaustive list includes [WLH19; Con+21; PSE21; LZW23; Ji+24]. Additionally, the model-based machine learning paradigm has been applied to this task in [Shm+23; Shm+25]. The MUSIC method considers several assumptions on the system model: e.g. sources must be non-coherent, the signals must be narrowband. In [Shm+23; Shm+25], the authors aimed to relax these assumptions by introducing SubspaceNet: a convolutional neural network that outputs a surrogate covariance matrix, that can then be used by the MUSIC method. Such method has been shown to counter the effect of coherent sources in addition to hardware impairments. The differentiable MUSIC (diff-MUSIC) method proposed in this chapter is aimed at achieving hardware impairments learning without the use of a CNN, thereby significantly reducing the number of learnable parameters compared to SubspaceNet.

Contributions

This chapter addresses the joint problem of DoA estimation and HWI learning within the model-based machine learning paradigm. Specifically, it introduces a differentiable MUSIC algorithm, thereby enabling the use of gradient-based optimization methods for the considered problem. The main contributions are as follows:

- A differentiable version of the classical MUSIC algorithm is introduced by replacing the non-differentiable arg max step by a convex combination of DoAs.
- Supervised and unsupervised learning strategies are presented, allowing an efficient learning process.
- Experiments conducted on synthetic data against several baselines demonstrate the effectiveness of the proposed method.

The method is shown to effectively compensate for significant antenna array impairments. In particular, this study considers impairments in both the location and complex gain of the radiating elements comprising the sensing antenna array.

7.1.3 Problem formulation

A uniform linear array composed of N antennas with half-wavelength spacing is considered [1]. $M < N$ non-coherent far-field sources are impinging the array. Measurements are carried out over T snapshots:

$$\mathbf{X} = \mathbf{A}_{\boldsymbol{\zeta}}\left(\boldsymbol{\theta}\right)\mathbf{S} + \mathbf{N}, \tag{7.1}$$

where $\mathbf{X} \in \mathbb{C}^{N\times T}$ corresponds to the measured signals, and $\boldsymbol{\theta} \in \left[-\pi/2, \pi/2\right]^M$ represents the sources DoAs. The narrowband sources signals are defined as $\mathbf{S} = \left[\mathbf{s}_1, \cdots, \mathbf{s}_T\right] \in \mathbb{C}^{M\times T}$, such that $\mathbf{s}_i \sim \mathcal{CN}\left(\mathbf{0}_{\mathbb{C}^M}, \sigma_s^2\mathbf{Id}_M\right)$. The sensing noise $\mathbf{N} = \left[\mathbf{n}_1, \cdots, \mathbf{n}_T\right] \in \mathbb{C}^{N\times T}$ is defined such that $\mathbf{n}_i \sim \mathcal{CN}\left(\mathbf{0}_{\mathbb{C}^N}, \sigma_n^2\mathbf{Id}_N\right)$. $\mathbf{A}_{\boldsymbol{\zeta}}\left(\boldsymbol{\theta}\right) = \left[\mathbf{a}_{\boldsymbol{\zeta}}\left(\theta_1\right), \cdots, \mathbf{a}_{\boldsymbol{\zeta}}\left(\theta_1\right)\right] \in \mathbb{C}^{N\times M}$ is a steering matrix defined with steering vectors:

$$\mathbf{a}_{\boldsymbol{\zeta}}\left(\boldsymbol{\theta}_i\right) = \frac{1}{\left\|\mathbf{g}\right\|_2}\begin{bmatrix} g_1 \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda}p_1 u(\theta_i)} \\ \vdots \\ g_N \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda}p_N u(\theta_i)} \end{bmatrix}, \tag{7.2}$$

where $\mathbf{g} = \left[g_1, \cdots, g_N\right]^\mathsf{T} \in \mathbb{C}^N$ represents the complex antenna gains, $p_i \in \mathbb{R}$ is the location of the ith antenna along the array axis, and $u\left(\theta_i\right) = \cos\left(\theta_i\right)$. Finally, $\boldsymbol{\zeta} = \left[\left\{g_i\right\}_{i=1}^N, \left\{p_i\right\}_{i=1}^N\right]$ corresponds to a physical parametrization of the array manifold. This system model is visualized in Fig. 7.1.

1. Note that the presented approach can be straightforwardly extended to any given antenna array geometry.

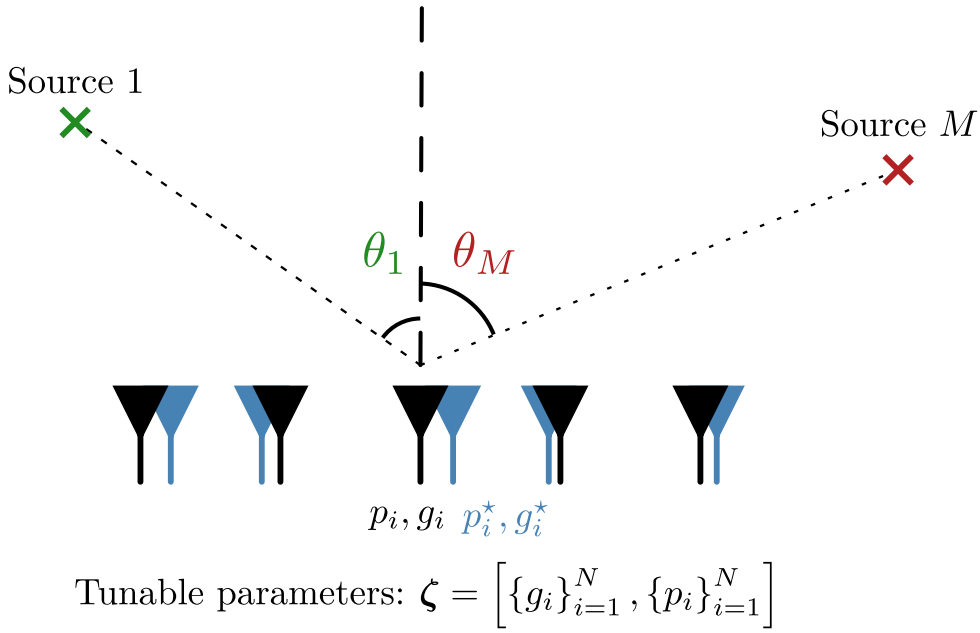


Figure 7.1: System model: blue antennas are the physical antennas, i.e. with HWI knowledge, black antennas represent the nominal antenna, i.e. without HWI knowledge.

Remark 7.1. *The system model presented in Eq.* (7.1)*, originates from (H1). Specifically, modeling the received signal as a noisy linear combination of steering vectors implicitly relies on (H1), since the steering vector model itself can be derived from it. This derivation constituted one of the key contributions presented in Chapter 4.*

The main inquiry of this chapter can be formulated as follows: *Can the DoAs* $\boldsymbol{\theta}$ *be recovered from the received signal* $\mathbf{X}$*, while efficiently learning the HWIs?*

7.2 Model-based approach

7.2.1 Subspace methods

The eigenstructure of the received signal covariance matrix can be exploited to perform DoA estimation. This methodology is referred to as subspace-based, as it involves the decomposition of the received signal into orthogonal components corresponding to the signal and noise subspaces. An overview of this subspace decomposition is presented in Highlight 7.1.

Highlight 7.1: Subspace decomposition

Given a received signal $\mathbf{X}$ following Eq. (7.1), and under the assumption of non-coherent sources, the covariance matrix of $\mathbf{X}$ can be expressed as:

$$\boldsymbol{\Gamma}_{\mathbf{X}} = \mathbf{A}_{\boldsymbol{\zeta}}(\boldsymbol{\theta})\,\boldsymbol{\Gamma}_{\mathbf{S}}\mathbf{A}_{\boldsymbol{\zeta}}^{\mathsf{H}}(\boldsymbol{\theta}) + \sigma_n^2\mathbf{Id}_N, \tag{7.3}$$

where $\boldsymbol{\Gamma}_{\mathbf{S}}$, the covariance matrix of $\mathbf{S}$, is diagonal. Since

$$\operatorname{rank}\left(\mathbf{A}_{\boldsymbol{\zeta}}(\boldsymbol{\theta})\,\boldsymbol{\Gamma}_{\mathbf{S}}\mathbf{A}_{\boldsymbol{\zeta}}^{\mathsf{H}}(\boldsymbol{\theta})\right) = M, \tag{7.4}$$

$\mathbf{A}_{\boldsymbol{\zeta}}(\boldsymbol{\theta})\,\boldsymbol{\Gamma}_{\mathbf{S}}\mathbf{A}_{\boldsymbol{\zeta}}^{\mathsf{H}}(\boldsymbol{\theta})$ admits only M strictly positive eigenvalues. Furthermore, $\boldsymbol{\Gamma}_{\mathbf{X}}$ admits the following eigenvalue decomposition (EVD):

$$\boldsymbol{\Gamma}_{\mathbf{X}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathsf{H}}, \tag{7.5}$$

where $\mathbf{U}$ is the matrix containing the eigenvectors and $\boldsymbol{\Lambda}$ is the diagonal matrix holding the eigenvalues on its diagonal. Then, combining Eq. (7.3) and (7.5) yields the following results [a]:

1. The eigenvalues ordered by decreasing amplitude are:

$$\lambda_1, \cdots, \lambda_M, \lambda_{M+1}, \cdots, \lambda_{M+1}, \tag{7.6}$$

   where $\lambda_{M+1} = \sigma_n^2$ has multiplicity $N - M$.

2. The eigenvectors span a space that can be decomposed into signal and noise subspaces:

$$\mathbf{U} = [\mathbf{U}_S, \mathbf{U}_N], \tag{7.7}$$

   with $\mathbf{U}_S \perp \mathbf{U}_N$.

As $\operatorname{span}(\mathbf{A}_{\boldsymbol{\zeta}}(\boldsymbol{\theta})) = \operatorname{span}(\mathbf{U}_S)$, it is then clearly established that $\mathbf{A}_{\boldsymbol{\zeta}}(\boldsymbol{\theta}) \perp \mathbf{U}_N$, leading to:

$$\forall i \in [\![1, M]\!], \left\|\mathbf{U}_N^{\mathsf{H}}\mathbf{A}_{\boldsymbol{\zeta}}(\theta_i)\right\|_2^2 = 0. \tag{7.8}$$

a. See [SM05, Chapter 4, pp.159-164], or Appendix F.1 for more details.

7.2.2 MUSIC

The core of the MUSIC algorithm lies in Eq. (7.8), and is summarized in Algorithm 5. Since the inverse of Eq. (7.8) tends to infinity for the sources' DoAs, the MUSIC algorithm involves evaluating this inverse on an angular grid $\boldsymbol{\theta}_g$ and identifying the sources' DoAs as

the arguments of the peaks. It is important to note from Eq. (7.13) that DoA estimation performance depends on the array parametrization knowledge: imperfect knowledge of the antenna gains and locations leads to error in the MUSIC spectrum peak locations and amplitudes, thereby altering the estimated DoAs, as presented in Fig. 7.2.

Algorithm 5 MUSIC algorithm.

Inputs: Measured signals $\mathbf{X} \in \mathbb{C}^{N \times T}$, number of sources M, angular grid $\boldsymbol{\theta}_g \in \mathbb{R}^{N_\theta}$, current array parametrization knowledge $\boldsymbol{\zeta}$.

1: Compute the sample covariance matrix:

$$\hat{\boldsymbol{\Gamma}}_{\mathbf{X}} = \frac{1}{T}\mathbf{X}\mathbf{X}^{\mathsf{H}}. \tag{7.9}$$

2: Perform the EVD of the sample covariance matrix:

$$\hat{\boldsymbol{\Gamma}}_{\mathbf{X}} = \tilde{\mathbf{U}}\tilde{\boldsymbol{\Lambda}}\tilde{\mathbf{U}}^{\mathsf{H}}. \tag{7.10}$$

3: Order the eigenvectors by decreasing eigenvalues amplitude, and perform subspace separation:

$$\tilde{\mathbf{U}} = \left[\tilde{\mathbf{U}}_S, \tilde{\mathbf{U}}_N\right]. \tag{7.11}$$

4: Compute the MUSIC spectrum along the angular grid:

$$P_{\text{MUSIC}}\left(\boldsymbol{\theta}_g|\boldsymbol{\zeta}\right) = \frac{1}{\left\|\tilde{\mathbf{U}}_N^{\mathsf{H}}\mathbf{A}_{\boldsymbol{\zeta}}\left(\boldsymbol{\theta}_g\right)\right\|_2^2}. \tag{7.12}$$

5: Estimate the DoAs as the argument of the M top-peaks:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}_g}{}_M P_{\text{MUSIC}}\left(\boldsymbol{\theta}_g|\boldsymbol{\zeta}\right). \tag{7.13}$$

Output: $\hat{\boldsymbol{\theta}} \in \mathbb{R}^M$ (Estimated DoAs)

7.3 Towards diffMUSIC

As previously mentioned, the performance of the classical MUSIC algorithm deteriorates when considering hardware impairments. To address this issue, it is proposed to modify its structure to make it differentiable, enabling the learning of the array parametrization within the DoA estimation scheme, through the use of first order methods.

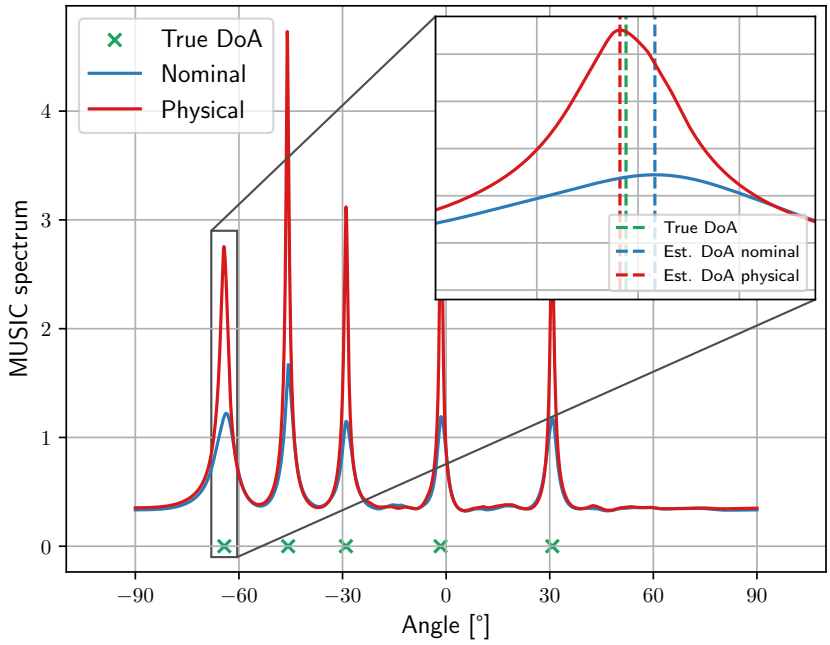


Figure 7.2: MUSIC performance under gain and location impairments: nominal, resp. physical, represents the spectrum without, resp. with, hardware impairment knowledge.

7.3.1 Model-based differentiable architecture

Stochastic gradient descent (SGD) is leveraged to solve:

$$\begin{aligned} \underset{\boldsymbol{\zeta}}{\text{minimize}} \quad & \mathbb{E}_{(\boldsymbol{\theta},\mathbf{X})\sim p_{\boldsymbol{\theta},\mathbf{X}}}\left[\mu\left(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}\left(\mathbf{X}|\boldsymbol{\zeta}\right)\right)\right], \\ \text{subject to} \quad & \boldsymbol{\zeta} \in \mathbb{C}^N \times \mathbb{R}^N \end{aligned} \tag{7.14}$$

where $p_{\boldsymbol{\theta},\mathbf{X}}$ represents the data distribution, μ is a loss function on the DoAs, $\boldsymbol{\theta}$ are the true DoAs associated to the measurements $\mathbf{X}$, and $\hat{\boldsymbol{\theta}}\left(\mathbf{X}|\boldsymbol{\zeta}\right)$ are estimated DoAs from a given DoA estimator with array parametrization knowledge $\boldsymbol{\zeta}$. In order to solve the problem described in Eq. (7.14) through SGD, it is necessary to compute the gradients $\nabla_{\zeta}\mu(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}\left(\mathbf{X}|\boldsymbol{\zeta}\right))$. Using the derivative chain rule [2], the following loss gradient expression can be obtained:

$$\nabla_{\zeta}\mu(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}\left(\mathbf{X}|\boldsymbol{\zeta}\right)) = \Delta^{\mathsf{T}}_{\hat{\boldsymbol{\theta}}(\mathbf{X}|\zeta),\zeta}\nabla_{\hat{\boldsymbol{\theta}}(\mathbf{X}|\zeta)}\mu\left(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}\left(\mathbf{X}|\boldsymbol{\zeta}\right)\right) \tag{7.15}$$

$$= \left[\nabla_{\zeta_1}\hat{\boldsymbol{\theta}}\left(\mathbf{X}|\boldsymbol{\zeta}\right),\cdots,\nabla_{\zeta_{2N}}\hat{\boldsymbol{\theta}}\left(\mathbf{X}|\boldsymbol{\zeta}\right)\right]^{\mathsf{T}}\nabla_{\hat{\boldsymbol{\theta}}(\mathbf{X}|\zeta)}\mu\left(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}\left(\mathbf{X}|\boldsymbol{\zeta}\right)\right). \tag{7.16}$$

It then becomes evident that evaluating the Jacobian $\Delta_{\hat{\boldsymbol{\theta}}(\mathbf{X}|\zeta),\zeta}$ is required in order to compute the gradients $\nabla_{\zeta}\mu(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}\left(\mathbf{X}|\boldsymbol{\zeta}\right))$: this defines the core challenge of this chapter,

2. See Definition 2.11.

as explained thereafter. The MUSIC algorithm is inherently non-differentiable, a characteristic arising from the arg max operation in Eq. (7.13). This hard decision scheme can be represented through angular masks containing a single value at each spectrum peak, with no operations performed within those windows. Since only the operations following the applications of the fixed angular masks are considered for gradient computations, it leads to the non-existence of $\Delta_{\hat{\boldsymbol{\theta}}(\mathbf{X}|\boldsymbol{\zeta}),\boldsymbol{\zeta}}$ for the MUSIC method, which results in its non-differentiability.

It is thus proposed to replace this hard-decision scheme with the differentiable softmax-based approach outlined in Algorithm 6, where each estimated DoA can be seen as a convex combination of grid DoAs around the associated MUSIC spectrum peak. Note that $\boldsymbol{\theta}_i^{\text{mask}} = \Pi_L(\boldsymbol{\theta}_g, \theta_i^{\text{peak}})$ denotes a windowing operation of size L, applied on the DoA grid $\boldsymbol{\theta}_g$, centered around θ_i^{peak} such that $\boldsymbol{\theta}_i^{\text{mask}} \in \mathbb{R}^L$. As for MUSIC, the diffMUSIC method initially estimates the spectrum peaks using a non-differentiable peak finding method: arg max in Eq. (7.17). Then, angular windows are defined around each peaks. As depicted in Eq. (7.19), within each window, the estimated DoAs are computed using differentiable operations involving the steering matrix $\mathbf{A}_{\boldsymbol{\zeta}}(\boldsymbol{\theta}_g)$. Indeed, the MUSIC spectrum is computed using this steering matrix. Since computations involving $\mathbf{A}_{\boldsymbol{\zeta}}(\boldsymbol{\theta}_g)$ are performed after the peek finding step, the existence of $\Delta_{\hat{\boldsymbol{\theta}}(\mathbf{X}|\boldsymbol{\zeta}),\boldsymbol{\zeta}}$ is guaranteed for a fixed angular mask, thereby making the method differentiable. Note that, in addition to being differentiable, this approach also increases angular resolution as off-grid estimates are possible through the convex DoA combination.

Algorithm 6 diffMUSIC algorithm.

Inputs: Computed MUSIC spectrum $P_{\text{MUSIC}}(\boldsymbol{\theta}_g|\boldsymbol{\zeta})$ from Eq. (7.12), angular window size L, current array parametrization knowledge $\boldsymbol{\zeta}$.

1: Find peaks in $P_{\text{MUSIC}}(\boldsymbol{\theta}_g|\boldsymbol{\zeta})$: $\boldsymbol{\theta}^{\text{peak}} = \left[\theta_1^{\text{peak}}, \cdots, \theta_M^{\text{peak}}\right]$:

$$\boldsymbol{\theta}^{\text{peak}} = \arg\max_{\boldsymbol{\theta}_g}{}_M P_{\text{MUSIC}}(\boldsymbol{\theta}_g|\boldsymbol{\zeta}). \tag{7.17}$$

2: **for each** θ_i^{peak} **do**

3: Compute an angular mask centered around θ_i^{peak}:

$$\boldsymbol{\theta}_i^{\text{mask}} = \Pi_L(\boldsymbol{\theta}_g, \theta_i^{\text{peak}}). \tag{7.18}$$

4: Estimate the associated DoA:

$$\hat{\theta}_i = \left(\boldsymbol{\theta}_i^{\text{mask}}\right)^{\mathsf{T}} \texttt{softmax}\left(P_{\text{MUSIC}}\left(\boldsymbol{\theta}_i^{\text{mask}}|\boldsymbol{\zeta}\right)\right). \tag{7.19}$$

5: **end for**

Output: $\hat{\boldsymbol{\theta}} \in \mathbb{R}^M$ (Estimated DoAs)

Table 7.1: Training complexity comparison: $M \ll N_\theta$.

	$\mu_{\mathrm{SL},\theta}$	$\mu_{\mathrm{SL},P}$	μ_{UL}
Time complexity	$\mathcal{O}\left(N^2 N_\theta + \kappa_{\mathrm{peak}} + \kappa_{\mathrm{EVD}} + LM\right)$	$\mathcal{O}\left(N^2 M + \kappa_{\mathrm{EVD}}\right)$	$\mathcal{O}\left(N^2 N_\theta + \kappa_{\mathrm{peak}}\right)$

7.3.2 Task-based loss functions

Supervised learning

As presented in [Shm+23; Shm+25], the root mean squared periodic error (RMSPE) on the DoAs can be used as a loss function to solve the optimization problem represented in Eq. (7.14), in a supervised learning (SL) manner. This loss function can be formally defined as:

$$\mu_{\mathrm{SL},\theta} = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{\theta},\mathbf{X})\in\mathcal{D}} \min_{\mathbf{P}\in\mathcal{P}} \frac{\left\|\mathrm{mod}_\pi\left(\boldsymbol{\theta} - \mathbf{P}\hat{\boldsymbol{\theta}}\left(\mathbf{X}|\boldsymbol{\zeta}\right)\right)\right\|_2}{\sqrt{M}}, \tag{7.20}$$

where $\mathcal{P}$ denotes the set of permutation matrices, and $\mathcal{D} = \{(\boldsymbol{\theta}_i, \mathbf{X}_i)\}_{i=1}^{N_t}$ is the training set. The RMSPE takes into account the permutation invariance in the DoA learning task through the permutation matrices $\mathbf{P} \in \mathcal{P}$ [3]. Rather than learning through estimated DoAs, a proposed alternative strategy is to maximize the MUSIC spectrum amplitude at the true DoA locations. The associated supervised loss is defined as:

$$\mu_{\mathrm{SL},P} = -\frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{\theta},\mathbf{X})\in\mathcal{D}} \sum_i P_{\mathrm{MUSIC}}\left(\theta_i|\boldsymbol{\zeta}\right). \tag{7.21}$$

This approach does not directly solve Eq. (7.14); instead, it addresses a proxy problem, the solution of which translates into minimizing the DoA estimation error. Indeed, minimizing $\mu_{\mathrm{SL},P}$ can be interpreted as finding an array parametrization that minimizes the correlation between steering vectors associated to the true DoAs and the noise subspace eigenvectors, i.e. it aims at maximizing orthogonality between signal and noise subspaces. As a matter of fact, it is shown in Appendix F.2, that minimizing $\mu_{\mathrm{SL},P}$ is equivalent to minimizing a principal-angle-based distance, which, when minimized, maximizes the distance between the signal and noise subspaces, thereby enforcing orthogonality.

Once the impairments are learned using $\mu_{\mathrm{SL},P}$, the DoAs can then be estimated using the classical MUSIC algorithm, or through diffMUSIC, with the learned array. This alternative SL strategy presents a reduced computational complexity during training in comparison to the RMSPE based strategy. This is presented in Table 7.1, where κ_{peak}, resp. κ_{EVD}, represents the time complexity associated with the peak finding function, resp. EVD. This complexity reduction arises from the fact that, with $\mu_{\mathrm{SL},P}$, the MUSIC spectrum only needs to be evaluated at the true DoA locations, rather than over a fine

3. More information about this metric can be found in [Shm+23].

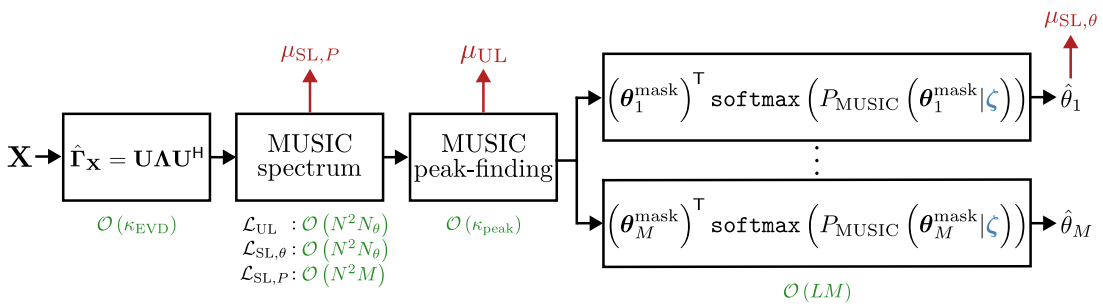


Figure 7.3: Illustration of the diffMUSIC method, in addition to the training complexity of the different loss functions: $M \ll N_\theta$.

angular grid, as required for computing the DoAs with $\mu_{\text{SL},\theta}$.

Unsupervised learning

The proposed SL approaches rely on true DoA knowledge, which may limit their practical applicability. An unsupervised learning (UL) strategy can be implemented by using a loss function designed to maximize the MUSIC spectrum peaks within each angular mask in diffMUSIC. Such loss function can, for instance, be based on the coefficient of variation, the Gini coefficient, the kurtosis or the Kullback-Leibler divergence relative to the uniform distribution. It is proposed to consider a Jain's index based loss function where minimization translates to a sharp peak inside the masked spectrum. The Jain's index is formally defined as follows:

$$\forall \mathbf{x} \in \mathbb{R}^n,\ \mathfrak{J}(\mathbf{x}) = \frac{\left(\sum_i x_i\right)^2}{n \sum_i x_i^2}, \tag{7.22}$$

and the associated unsupervised loss function can be defined as:

$$\mu_{\text{UL}} = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{X} \in \mathcal{D}} \sum_i \mathfrak{J}\left(P_{\text{MUSIC}}\left(\boldsymbol{\theta}_i^{\text{mask}}(\mathbf{X}|\boldsymbol{\zeta})\,|\boldsymbol{\zeta}\right)\right), \tag{7.23}$$

where $\boldsymbol{\theta}_i^{\text{mask}}(\mathbf{X}|\boldsymbol{\zeta})$ is the obtained angular mask for given measurements in Eq. (7.18). In addition to Table 7.1, Fig. 7.3 presents the overall diffMUSIC structure, as well as the computational complexity of the different proposed loss functions.

7.4 Experiments

It is proposed to study the impairments learning performance of the proposed diffMUSIC method in different scenarios against several baselines.

7.4.1 Learning framework and baselines

Data generation

A ULA with $N = 16$ antennas is considered, and measurements are generated following Eq. (7.1). The sources DoAs are generated as:

$$\forall i \in [\![1, M]\!], \theta_i \sim \mathcal{U}\left[-80°, 80°\right]. \tag{7.24}$$

The sensing SNR is computed as $\mathsf{SNR} = 10\log_{10}\left(\sigma_s^2/\sigma_n^2\right)$. Concerning the hardware impairments, imperfections in both antenna locations and gains are taken into account. For the antenna locations, one has:

$$\forall i \in [\![1, M]\!], p_i = \tilde{p}_i + \delta_{p_i}, \tag{7.25}$$

where p_i, resp. $\tilde{p}_i$, denotes the physical, resp. nominal, ith antenna location, and $\delta_{p_i} \sim \mathcal{U}\left[-\eta, \eta\right]$, with $\eta \propto \lambda/2$ [4]. Similarly, for the antenna gains, one has:

$$\forall i \in [\![1, M]\!], g_i = \tilde{g}_i + \delta_{g_i}, \tag{7.26}$$

where g_i, resp. $\tilde{g}_i$, denotes the physical, resp. nominal, complex gain for the ith antenna, and $\delta_{g_i} \sim \mathcal{CN}\left(0, \sigma_g^2\right)$.

Unless otherwise stated, $M = 5$ sources, $T = 100$ snapshots, $\eta = 0.5\lambda/2$, and $\sigma_g^2 = 0.36$ are considered.

Baselines

The proposed approach is compared against the classical MUSIC algorithm, denoted as `M`, with nominal and physical array, but also against diffMUSIC, denoted as `dM`, with physical array. Note that the window size L of diffMUSIC is optimized in each considered scenario. Additionally, it is proposed to compare diffMUSIC against the SubspaceNet network presented in [Shm+23]: as its goal is to learn a surrogate covariance matrix to counter imperfections in the sensing process, it is able to take into account hardware impairments. The evaluation metric for all approaches is the RMSPE defined in Eq. (7.20), evaluated over a test set.

7.4.2 Results

Performance against baselines.

Table 7.2 presents a DoA estimation performance comparison for $M \in \{1, 5\}$ sources, at 30dB SNR. It can be seen that the proposed approaches, i.e. `dM` with the different loss functions, outperform the MUSIC algorithm with nominal array knowledge. Moreover,

4. Note that the locations are only altered along the non-null dimension of the ULA.

one can see that SubspaceNet performs relatively well for $M = 1$ but totally fails to learn the imperfections in the multi-source scenario. Additionally, it can be seen that diffMUSIC with $\mu_{\mathrm{SL},P}$ equals or outperforms diffMUSIC with physical array, indicating good impairment learning capabilities. Furthermore, one can remark that the UL approach presents good performance in the multi-sources scenario: this is of particular interest as this approach does not rely on labeled DoAs. Note that this approach could be extended in an online learning strategy where the antenna array would learn its hardware impairments on the fly. Finally, the superior performance of diffMUSIC compared to MUSIC, both evaluated using a physical array, highlights the off-grid estimation capabilities of the diffMUSIC method.

Learned parameters and MUSIC spectrums

It is proposed to visualize both the learned parameters by the proposed methods for $M = 5$, and their impact on the associated MUSIC spectrums.

Fig. 7.4 presents the learned parameters: the center of each circle represents an antenna location, the circle radius represents the associated complex gain magnitude, and the segment angle represents the associated complex gain phase. It can be seen that the $\mu_{\mathrm{SL},\theta}$ approach presents good impairments learning except for two antennas whose defaults are small. It is worth noting that both the $\mu_{\mathrm{SL},P}$ and μ_{UL} approaches achieve near-perfect learning of the impairments which aligns with their strong performance observed in Table 7.2.

Fig. 7.5 presents the spectrums obtained with the learned parameters by the different approaches. Results are illustrated at 10dB SNR for better visualization. One can remark that, as expected by the definition of the objective of this loss function, the spectrum obtained with the array parametrization learned through $\mu_{\mathrm{SL},P}$ tightly follows the physical spectrum. One can also note that the peaks of the spectrum obtained through $\mu_{\mathrm{SL},\theta}$ are aligned with the ones of the physical spectrum. Unsurprisingly, an angular shift is observed in the peaks of the spectrum obtained through μ_{UL}: while this approach allows to maximize peak amplitudes, its unsupervised nature does not allow to introduce inductive angular bias on the peak locations during training. This angular shift phenomenon is particularly noticeable near the endfire of the antenna array.

Performance against sensing noise

Fig. 7.6 presents the DoA estimation performance evolution with variable SNR, for $M = 5$ sources and $T = 100$ snapshots. One can remark that, while the $\mu_{\mathrm{SL},\theta}$ approach presents good performance in the high SNR region, the $\mu_{\mathrm{SL},P}$ approach presents the overall best performance as it follows tightly the diffMUSIC performance with physical array. It is worth noting that, in the very low SNR region, both the $\mu_{\mathrm{SL},\theta}$ and μ_{UL} approaches are significantly affected by noise, preventing them from effectively learning hardware impairments. In contrast, the $\mu_{\mathrm{SL},P}$ approach does not suffer from that issue.

Table 7.2: Baseline comparisons: `M` corresponds to the classical MUSIC algorithmic, while `dM` correspond to the proposed `dM` approach. For the $\mu_{\mathrm{SL},\theta}$, $\mu_{\mathrm{SL},P}$, and μ_{UL} columns, `M` denotes the results when using MUSIC during inference, but diffMUSIC during training.

		Baselines				$\mu_{\mathrm{SL},\theta}$		$\mu_{\mathrm{SL},P}$		μ_{UL}	
		`M` (nom.)	`M` (phys.)	`dM` (phys.)	`SubspaceNet`	`M`	`dM`	`M`	`dM`	`M`	`dM`
RMSPE [°]	$M = 1$	2.425	0.014	0.013	0.098	0.019	0.015	0.013	**0.013**	1.339	1.310
	$M = 5$	9.976	4.358	4.275	16.123	5.371	5.178	4.325	**4.209**	4.834	4.731

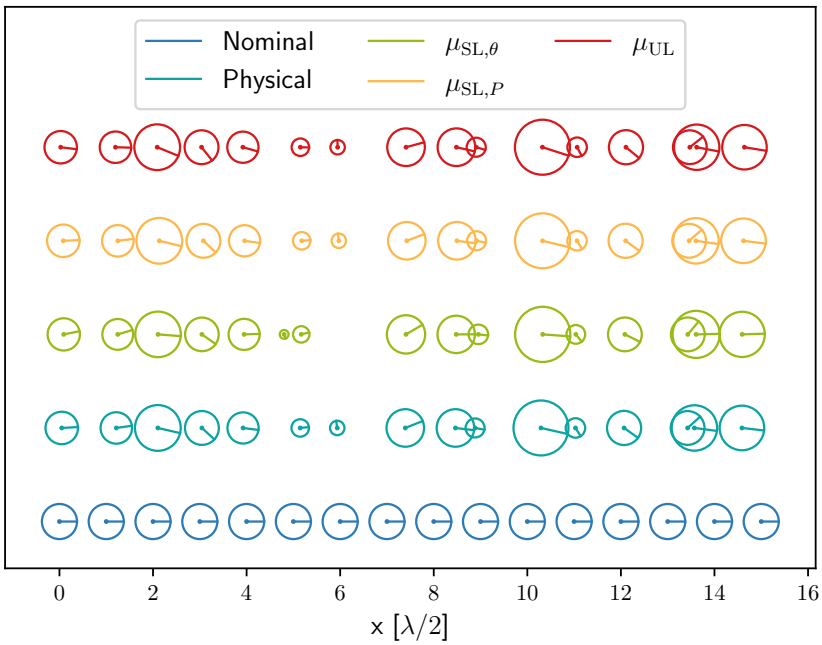


Figure 7.4: Learned parameters comparison for $M = 5$, 30dB SNR.

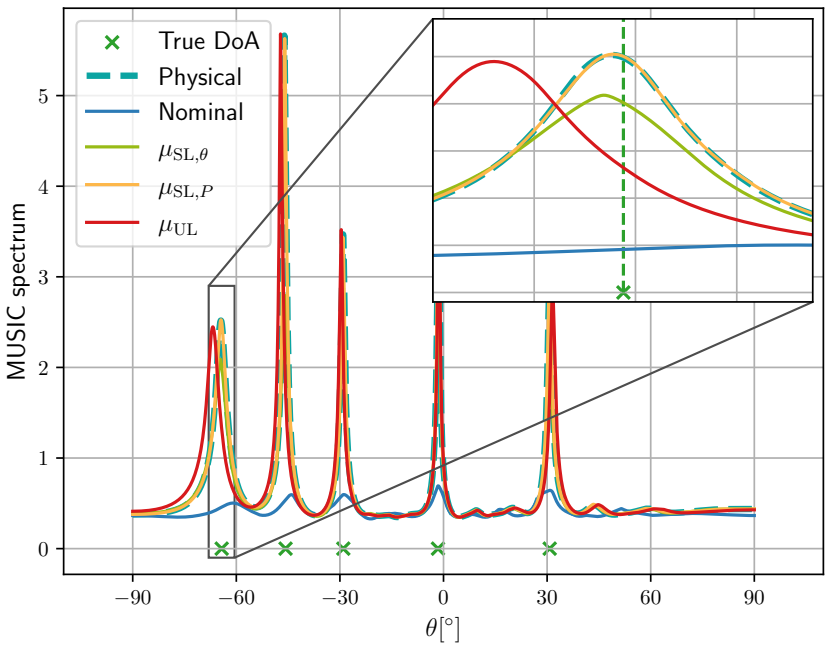


Figure 7.5: MUSIC spectrums with learned arrays, $M = 5$, $T = 100$, 10dB SNR.

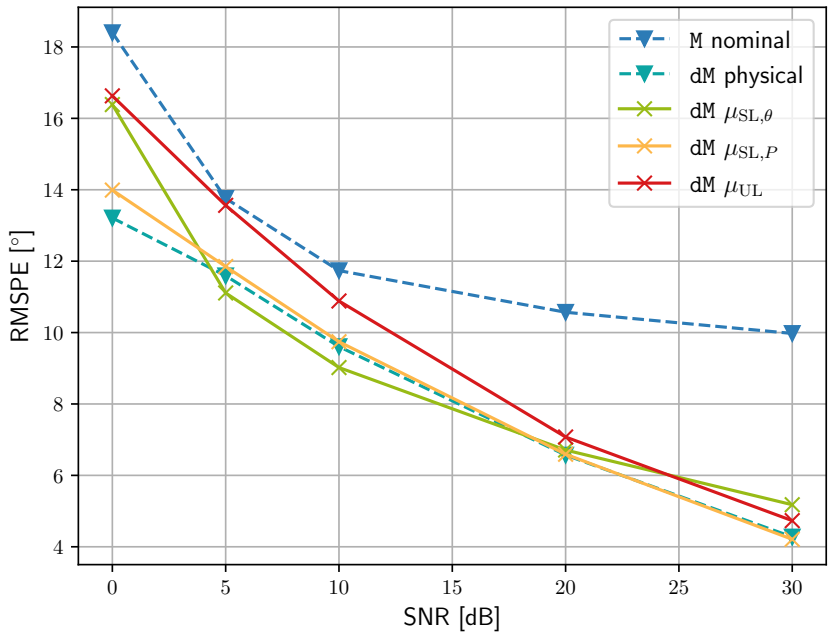


Figure 7.6: RMSPE evolution with sensing SNR, $T = 100$.

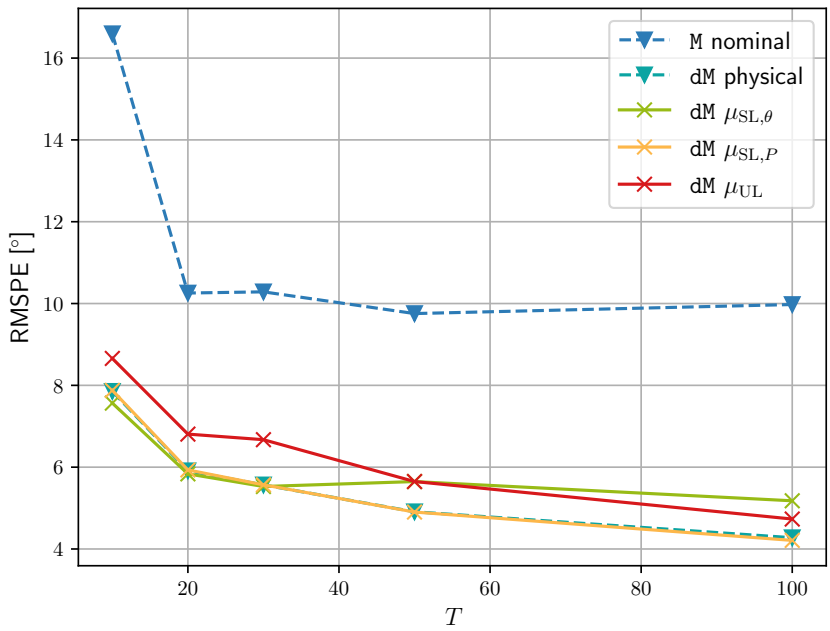


Figure 7.7: RMSPE evolution with snapshot number, 30dB SNR.

Performance against snapshot number

Fig. 7.7 presents the DoA estimation performance evolution with variable number of snapshots for $M = 5$ sources and 30dB SNR. As expected, the estimation performance degrades across all approaches when the number of snapshots is low. This can be attributed to the higher estimation error in the empirical covariance matrix when T is low. It is also worth noting that the $\mu_{\mathrm{SL},P}$ approach exhibits the overall best performance. It demonstrates its superiority against other methods as it offers both better estimation performance in low SNR or low snapshot scenarios, while also achieving lower time complexity.

7.5 Conclusion

The objective of this chapter was to apply the MB-ML paradigm to the joint estimation of HWIs and DoAs, as formalized in Problem 3.3. Several remarks can be made regarding this approach:

- Following the MB-ML paradigm, the problem's domain knowledge, formalized in hypothesis (H1), was leveraged to propose an analytical model of the received signal. This model was subsequently used to **structure** a learning method, in the form of a differentiable algorithm, based on the classical MUSIC algorithm. Domain knowledge was also used to **initialize** the learning method, by initializing the differentiable MUSIC algorithm with a dictionary of nominal steering vectors.

- Multiple experiments demonstrated the strong learning performance of the proposed method. In particular, it was shown to accurately recover strong HWIs, both on complex gains and positions of the antenna array elements, even in the presence of high sensing noise and low number of measurement snapshots. Moreover, the proposed differentiable peak-finding mechanism enables angular estimates to be obtained in a continuous domain rather than being restricted to a fixed angular grid, thereby enhancing the overall estimation resolution.

- Several task-specific loss functions were introduced. The proposed unsupervised learning loss is especially noteworthy, as it enables hardware impairment compensation without requiring ground-truth DoA measurements, while also reducing computational complexity compared to the classical RMSPE-based loss.

Chapter 8

Channel charting and beam prediction

This chapter focuses on designing an efficient beam prediction framework in a cell-free massive MIMO system. The proposed approach leverages the model-based machine learning paradigm to develop both a neural encoder and decoder. The encoder builds on the channel charting concept, which is introduced in Section 8.1. Subsequently, Section 8.2 presents a formal definition of the considered problem. Section 8.3 details the proposed neural network architectures and the overall beam prediction framework. Experimental results validating the effectiveness of the proposed approach are presented in Section 8.4. Finally, Section 8.5 provides some concluding remarks.

Contents

8.1 Primer on channel charting **206**
- 8.1.1 Problem formulation 206
- 8.1.2 Performance measures 207
- 8.1.3 Towards a model-based approach 209

8.2 Problem setting **211**
- 8.2.1 General introduction 211
- 8.2.2 Related work and contributions 212
- 8.2.3 Problem formulation 213

8.3 Proposed method **215**
- 8.3.1 Inference 215
- 8.3.2 Training 217

8.4 Experiments **218**
- 8.4.1 Learning framework and baselines 218
- 8.4.2 Results 219

8.5 Conclusion **224**

This chapter is mainly based on a conference article presented at the *2023 57th Asilomar Conference on Signals, Systems, and Computers* [Yas+23]. Furthermore, the experiments described in this chapter entailed a total computational cost of approximately 3 GPU days.

8.1 Primer on channel charting

Channel charting, introduced in [Stu+18], is a powerful tool for dimensionality reduction problems in wireless communications, and serves as the core of the contributions presented in Chapters 8 and 9. As such, this section aims at providing a brief introduction to the CC concept.

8.1.1 Problem formulation

From a high level perspective, channel charting is an unsupervised dimensionality reduction technique that aims at embedding propagation channels in a low-dimensional representation space. The embedded channels are usually referred to as *pseudo-locations*, *channel charts*, or *charts*. The core of the CC concept lies in the preservation of spatial neighborhoods between the channel and pseudo-location spaces.

Highlight 8.1: Channel charting

Let $\mathcal{S} \subset \mathbb{R}^3$ define a location space, in which a BS equipped with N_a antennas operates on N_s distinct subcarriers. Given N different UEs located at $\mathbf{x}_i \in \mathcal{S}$, their channels are denoted as $\mathbf{h}(\mathbf{x}_i) \in \mathbb{C}^{N_a N_s}$. Channel charting consists in finding an encoding function:

$$\begin{aligned} \mathsf{E}\colon \mathbb{C}^{N_a N_s} &\longrightarrow \mathcal{Z} \\ \mathbf{h}(\mathbf{x}) &\longrightarrow \mathbf{z}, \end{aligned} \tag{8.1}$$

where $\mathcal{Z}$ is an embedding space such that:

$$\dim_{\mathbb{R}}(\mathcal{Z}) \ll \dim_{\mathbb{R}}\left(\mathbb{C}^{N_a N_s}\right). \tag{8.2}$$

Additionally, the core of the CC concept consists in finding an encoding function for which spatial neighborhoods are preserved between the channel and representation spaces, i.e. $\forall (i,j), i \neq j$:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 \simeq \gamma \|\mathbf{z}_i - \mathbf{z}_j\|_2, \tag{8.3}$$

where $\gamma \in \mathbb{R}$ is a scaling constant, and $\mathbf{z}_k = \mathsf{E}(\mathbf{h}(\mathbf{x}_k))$, for $k \in \{i,j\}$.

An illustration of the CC concept is provided in Fig. 8.1. For both (a) and (b), UE locations are color-coded, and the corresponding pseudo-locations, computed from the associated channels, are represented using the same colors. This visualization reveals that, from a high level perspective, spatial neighborhoods, i.e. cluster of similarly colored points, are largely preserved through the encoding function. However, one can remark that the charting procedure can create false neighborhoods in the representation space. For instance,

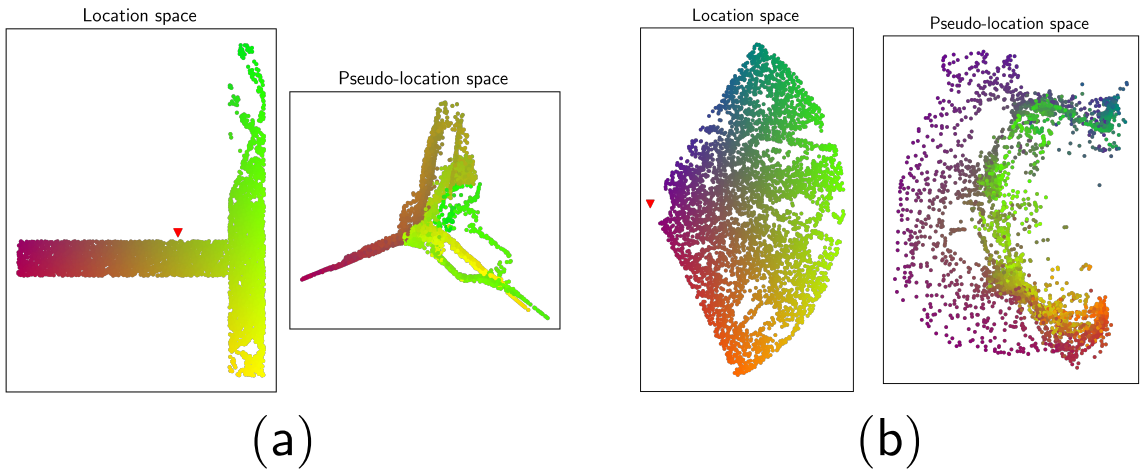


Figure 8.1: Illustration of the channel charting concept: (a) and (b) represents two different propagation environments, where the red triangle represents the BS location. In the used charting procedure, $\dim_{\mathbb{R}}(\mathcal{Z}) = 5$, but only the first two dimensions of the pseudo-locations are represented.

in Fig. 8.1 (a), green and yellow points appear close in the pseudo-location space but are distant in the location space. Moreover, while this graphical representation provides qualitative insights into the behavior of the channel charting method, it remains insufficient for a rigorous assessment of its performance. This highlights the necessity of quantitative performance metrics.

8.1.2 Performance measures

Within the CC framework, the quality of the used encoding function is evaluated based on its ability to preserve topological structures, such as spatial neighborhoods or global structures, between the channel and representation spaces. Standard quantitative metrics consists of the continuity (CT), trustworthiness (TW) [VK01], and Kruskal's stress (KS) [Kru64].

Continuity

The CT quantifies how much the encoding function preserves the spatial neighborhoods of the location space, that is how much channels corresponding to close locations are embedded in pseudo-locations that are also close. A formal definition is presented in Definition 8.1.

Definition 8.1: Continuity

Let $\mathcal{N}_K^{\mathcal{S}}(\mathbf{x}_i)$ denote the set of indices of the K-nearest neighbors of $\mathbf{x}_i$ in the location space $\mathcal{S}$. The CT is defined as:

$$\mathrm{CT}(K) = 1 - \frac{2}{NK(2N-3K-1)} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_K^{\mathcal{S}}(\mathbf{x}_i)} \hat{r}(i,j) - K, \tag{8.4}$$

where $\hat{r}(i,j)$ denotes the rank of $\mathbf{z}_j$ relative to $\mathbf{z}_i$, in terms of Euclidean distance in $\mathcal{Z}$, i.e. $\hat{r}(i,j) = n$ if $\mathbf{z}_j$ is the nth closest pseudo-location to $\mathbf{z}_i$ among all N pseudo-locations.

This metric is within the $[0,1]$ range, 1 meaning that for all of the N locations, their K-nearest neighbors are embedded among the K-nearest neighbors of the associated pseudo-locations, that is that all K-neighborhoods are preserved through the dimensionality reduction method. On the contrary, a zero-valued CT means that the embedding did not preserve any K-neighborhood.

Trustworthiness

The TW can be seen as the dual of the CT, i.e. how much spatial neighborhoods in the pseudo-location space correspond to actual neighborhoods in the location space. This metric can therefore be interpreted as an indicator to which extent the encoding function avoids creating false neighborhoods in the pseudo-location space. A formal definition is presented in Definition 8.2.

Definition 8.2: Trustworthiness

Let $\tilde{\mathcal{N}}_K^{\mathcal{Z}}(\mathbf{z}_i)$ denote the set of indices of the K-nearest neighbors of $\mathbf{z}_i$ in the pseudo-location space $\mathcal{Z}$, which are not its neighbors in the location space $\mathcal{S}$. The TW is defined as:

$$\mathrm{CT}(K) = 1 - \frac{2}{NK(2N-3K-1)} \sum_{i=1}^{N} \sum_{j \in \tilde{\mathcal{N}}_K^{\mathcal{Z}}(\mathbf{z}_i)} r(i,j) - K, \tag{8.5}$$

where $r(i,j)$ denotes the rank of $\mathbf{x}_j$ relative to $\mathbf{x}_i$ in terms of Euclidean distance distance in $\mathcal{S}$.

Similarly to the CT, the TW lies within the $[0,1]$ range, and is optimal when one-valued.

Kruskal's stress

Rather than quantifying how well spatial neighborhoods are preserved, KS quantifies how well the global structures in the location space are preserved in the pseudo-location space. A formal definition is proposed in Definition 8.3.

Definition 8.3: Kruskal's stress

The KS is defined as:

$$\mathrm{KS} = \sqrt{\frac{\sum_{i,j}\left(\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \gamma\,\|\mathbf{z}_i - \mathbf{z}_j\|_2\right)^2}{\sum_{i,j}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}}, \tag{8.6}$$

where the weighting term γ is defined as:

$$\gamma = \frac{\sum_{i,j}\|\mathbf{x}_i - \mathbf{x}_j\|_2\,\|\mathbf{z}_i - \mathbf{z}_j\|_2}{\sum_{i,j}\|\mathbf{z}_i - \mathbf{z}_j\|_2^2}. \tag{8.7}$$

In opposition to the CT and TW metrics, the KS is optimal when zero-valued.

8.1.3 Towards a model-based approach

As previously outlined, the CC concept is based on the use of an encoding function. There exists numerous approaches to design this function, some of which includes feature selection [Stu+18], siamese networks [Lei+19] or triplet networks [Fer+21; Yas+22].

In this thesis, a particular focus is brought to the use of the `ISOMAP` algorithm, introduced in [TdSL00], which allows dimensionality reduction through a process involving the computation of a distance graph. Owing to this distance reliance, it is necessary to design a distance that is well adapted to the charting problematics. Specifically, the study of Le Magoarou in [LeMag21a] showed the inadequacy of the classical Euclidean distance in this context: the main findings are summarized in Highlight 8.2.

Highlight 8.2: [LeMag21a] Euclidean distance and channel charting

Considering the Euclidean distance for CC, i.e.:

$$d\left(\mathbf{h}_i, \mathbf{h}_j\right) = \|\mathbf{h}_i - \mathbf{h}_j\|_2^2, \tag{8.8}$$

several remarks can be made, as listed below.

1. **Path loss impact.** Due to path loss, channels corresponding to locations

that are far from the base stations possess a small Euclidean distance, thereby making their pseudo-locations close. Conversely, channels corresponding to locations that are near the base station appear spaced in the representation space.

2. **Normalization impact.** A natural solution to the aforementioned problem consists in normalizing the channels, i.e. considering:

$$d\left(\mathbf{h}_i, \mathbf{h}_j\right) = \left\| \frac{\mathbf{h}_i}{\|\mathbf{h}_i\|_2} - \frac{\mathbf{h}_j}{\|\mathbf{h}_j\|_2} \right\|_2^2. \tag{8.9}$$

However, using this distance, a pathological case arises: when UEs are in the same direction to the BS, but spaced by half a wavelength. In this case, it can be shown that the normalized Euclidean distance is maximal, yielding maximally spaced channels, and thus pseudo-locations, for UEs that are usually spaced by only a few centimeters.

These two limitations are illustrated in Fig. 8.2.

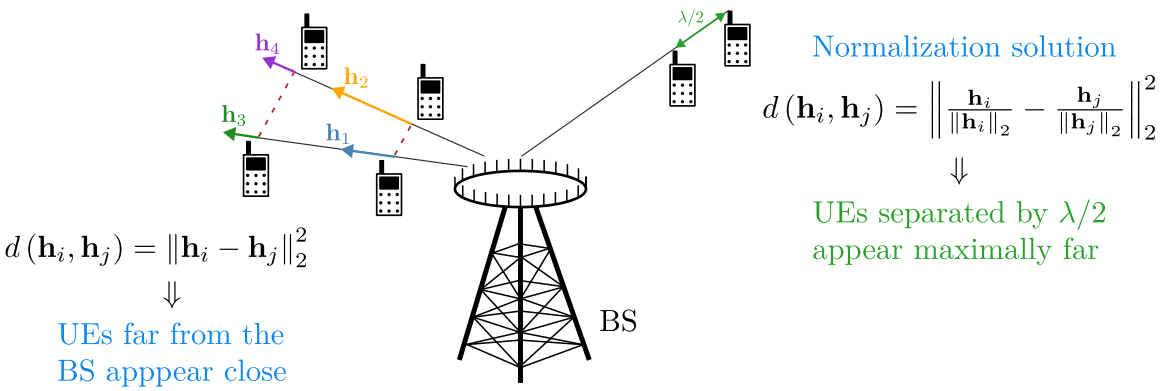


Figure 8.2: Illustration of Highlight 8.2.

Due to the inherent limitations of the Euclidean distance for CC, an adapted *phase-insensitive* (PI) distance was introduced in [LeMag21a]. It is defined as the solution of an optimization problem, which admits a closed-form solution:

$$d\left(\mathbf{h}_i, \mathbf{h}_j\right) = \min_{\phi \in [0, 2\pi]} \left\| \frac{\mathbf{h}_i}{\|\mathbf{h}_i\|_2} - \mathrm{e}^{\mathrm{j}\phi} \frac{\mathbf{h}_j}{\|\mathbf{h}_j\|_2} \right\|_2^2 \tag{8.10}$$

$$= \sqrt{2 - 2\frac{\left|\mathbf{h}_i^{\mathsf{H}} \mathbf{h}_j\right|}{\|\mathbf{h}_i\|_2 \|\mathbf{h}_j\|_2}}. \tag{8.11}$$

This approach originates from the observation that the global phase of channels varies

rapidly with the UE location, leading to the undesired effects described in Highlight 8.2. Therefore, introducing a phasor to align the channels allows to compensate for this effect.

Remark 8.1. *All theoretical developments leading to the observations presented in Highlight 8.2 and in the PI distance rely on the assumption that channels are pure steering/frequency response vectors, which can be traced back to (H1), as presented in Chapter 4.*

Remark 8.2. *Note that the use of the PI distance relies on perfect synchronization between the BS and each UE. A UE-dependent time offset Δt_i biases the perceived distances at the BS, leading to degraded CC performance. Indeed, considering UEs located at the same distance d from the BS, and thus sharing the same LoS propagation delay τ, their frequency response vectors, corresponding to the LoS channel, can be expressed, for UE i, as:*

$$\psi_{f,i}(\tau) = \begin{bmatrix} \mathrm{e}^{-\mathrm{j}2\pi f_1(\tau+\Delta t_i)} \\ \vdots \\ \mathrm{e}^{-\mathrm{j}2\pi f_{N_s}(\tau+\Delta t_i)} \end{bmatrix}, \tag{8.12}$$

where N_s denotes the number of considered frequencies. It becomes evident that, in the presence of unsynchronized UEs, i.e. when each UE has a specific time offset Δt_i, the corresponding FRVs become UE-dependent, whereas they should ideally be identical, since all UEs share the same distance from the BS.

While multiple distances can be used as channel dissimilarity metrics [1], the contributions presented in this chapter and the following exclusively rely on the PI distance.

8.2 Problem setting

8.2.1 General introduction

Large amounts of available bandwidth in the millimeter wave (mmWave) frequency bands is a key enabler of the high throughput requirements of 5G and Beyond 5G communication systems. Such high frequencies allow antenna sizes to decrease, resulting in practical mMIMO systems. As such systems possess high spatial directivity, it is of paramount importance to have precise beam management methods to benefit from the associated beamforming gain. Indeed, in mMIMO systems, a small angular error in the BSs beam steering angle towards the UE of interest could lead to a highly degraded link performance [CC22b]. In order to keep beam selection tractable, 3GPP standardized beam management procedures for 5G New Radio (5GNR) [3GP23]. The standard imposes for each BS to possess a codebook of precoders from which it picks the one that maximizes link performance for each individual UE. This procedure has to be carried out frequently as the UEs are likely to be moving. All of those constraints result in a complex beam management procedure, requiring constant exchanges between all BSs and UEs.

1. An interested reader is referred to [SEB24, Fig. 2] for further details.

In the last decade, machine learning (ML) has emerged as a promising solution in many communication problems such as channel estimation [He+18; BDA20; YLM22; CLMR23; Cha+24b], detection [SDW19; Cor+18], positioning [Cha+23] or decoding [Cor+22]. ML methods have been successfully applied for location-based beamforming [LeMag+22b] and beam management [Pon+21; Kaz+23]. More particularly, approaches combining channel charting and deep learning have been used for beam prediction and precoder learning in non-standalone mmWave [Pon+21] and cell-free (CF) mMIMO systems [LeMag+22a].

In CF mMIMO systems, a UE can be served by multiple BSs. In such scenarios, one can think of the following approach: through channel charting, a BS generates a pseudo-location from a collected uplink channel, and sends this pseudo-location to other BSs. Then, at each BS, a neural network is used to infer the best beam in its codebook, based solely on the received pseudo-location. This approach only requires estimating the uplink channel at one BS, resulting in a significant reduction of beam management complexity.

8.2.2 Related work and contributions

Contributions

This chapter tackles the task of jointly encoding and decoding CSI into a beam index, as formalized in Problem 3.4. While the use of the model-based approach for this encoding/decoding task has been previously investigated in [LeMag+22a], this chapter considers a similar methodology in a different setting, namely using codebooks rather than directly decoding precoders. The key contributions of this chapter are summarized as follows.

- A model-based neural encoder and decoder are considered to jointly encode a channel into a low-dimensional representation, and decode it into a best beam index in a codebook.
- The performance of the proposed encoder/decoder approach are assessed on realistic synthetic data against classical approaches, highlighting its performance.
- The proposed method, that can be seen as solving a classification task, is also compared to a regression approach, where a precoder is directly learned without constraining it to come from a given codebook.

Overall, the proposed approach is shown to greatly reduce the beam management complexity compared to classical methods, while presenting satisfactory beamforming performance.

Related work

Inferring a beam in a codebook from a pseudo-location has recently been proposed in [Pon+21]. However, this paper dit not followed the MB-ML paradigm for the decoding network, as it used a classical MLP to predict beams. Additionally, the performance of the proposed method were only evaluated on a small scene. Directly learning

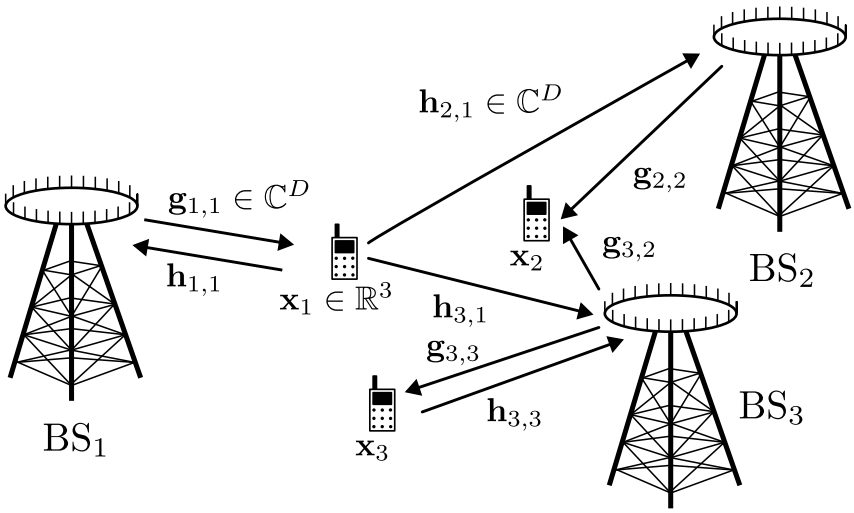


Figure 8.3: System model: here $B = 3, N_u = 3$.

a precoder has also been studied in [LeMag+22b; LeMag+22a; MEE10]. More particularly, [LeMag+22a] proposed a model-based neural network for the precoder learning task. However in [LeMag+22a], directly learned precoders were not compared to a codebook-based classification approach.

8.2.3 Problem formulation

In this chapter, a CF mMIMO system is considered, as depicted on Fig. 8.3, comprising B BSs using uniform planar arrays (UPAs) with N_a antennas, and N_u single-antenna UEs. Multicarrier transmissions with N_s subcarriers are considered. Note that the approach could be easily transposed with multi-antenna UEs as in [Pon+21]. The system operates with different uplink and downlink central frequencies f_{ul} and f_{dl}. Let $\mathbf{h}_{i,j} \in \mathbb{C}^D$, resp. $\mathbf{g}_{i,j} \in \mathbb{C}^D$, be the uplink, resp. downlink, channel between BS i and UE j. Note that vectorized channels are considered, hence $D = N_a N_s$. The location of UE j is denoted as $\mathbf{x}_j \in \mathbb{R}^3$.

Each BS uses the same 2D Discrete Fourier Transform (DFT) codebook $\mathbf{C} \in \mathbb{C}^{N_b \times N_a}$. It is defined using the Kronecker product as follows:

$$\mathbf{C} = \frac{1}{\sqrt{N_v N_h}} \mathbf{\Psi}_v \otimes \mathbf{\Psi}_h, \tag{8.13}$$

where $\mathbf{\Psi}_v \in \mathbb{C}^{O_v N_v \times N_v}$ and $\mathbf{\Psi}_h \in \mathbb{C}^{O_h N_h \times N_h}$, with N_h and O_h, resp. N_v and O_v, representing the number of antennas and oversampling factor for the horizontal, resp. vertical, dimension of the BS UPA. Note that the number of beams in the 2D-DFT codebook is a multiple of the number of BS antennas: $N_b = O_v O_h N_v N_h = O_v O_h N_a$, as $N_a = N_v N_h$. In this chapter, $O_h = O_v = 2$ so that the oversampling factor is uniformly split between the azimuth and elevation dimensions. For $k \in \{h, v\}$, the 1D-DFT codebook can be defined

as follows:

$$\mathbf{\Psi}_k = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \mathrm{e}^{\mathrm{j}2\pi\frac{1}{O_k N_k}} & \cdots & \mathrm{e}^{\mathrm{j}2\pi\frac{(N_k-1)}{O_k N_k}} \\ \vdots & & & \\ 1 & \mathrm{e}^{\mathrm{j}2\pi\frac{(O_k N_k-1)}{O_k N_k}} & \cdots & \mathrm{e}^{\mathrm{j}2\pi\frac{(O_k N_k-1)(N_k-1)}{O_k N_k}} \end{bmatrix}. \tag{8.14}$$

For each UE u, the best beam in the codebook at BS b is $\mathbf{c}_{i^\star_{b,u}} \in \mathbb{C}^{N_a}$, with:

$$i^\star_{b,u} = \arg\max_i \frac{1}{N_s} \sum_{s=1}^{N_s} \frac{\left|\mathbf{c}_i^{\mathsf{H}} \tilde{\mathbf{g}}_{b,u,s}\right|^2}{\|\tilde{\mathbf{g}}_{b,u,s}\|_2^2}, \tag{8.15}$$

where $\tilde{\mathbf{g}}_{b,u,s} \in \mathbb{C}^{N_a}$ is the downlink channel between BS b and UE u at subcarrier s. In other words, the chosen precoder maximizes the mean correlation over all subcarriers between the precoder and the downlink channel. The kth best beam index at BS b for UE u is denoted by $i^{\star,k}_{b,u}$.

Objective

The pursued objective is to choose an appropriate precoder based on a low-dimensional representation of the channel, possibly computed at another BS and/or at a different frequency. To do so, channel charting is used to obtain a pseudo-location $\mathbf{z} \in \mathbb{R}^d$ from an uplink channel $\mathbf{h} \in \mathbb{R}^D$ estimated at one BS. That pseudo-location is then sent to all the other BSs, which can subsequently infer the best beam in their respective codebook based on that pseudo-location. Formally, the proposed contribution can be defined as an encoding function whose role is to compress channels, via channel charting:

$$\begin{aligned} \mathsf{E}\colon \mathbb{C}^D &\longrightarrow \mathbb{R}^d \\ \mathbf{h} &\longrightarrow \mathbf{z} = \mathsf{E}\left(\mathbf{h}\right), \end{aligned} \tag{8.16}$$

and a decoding function whose role is to map compressed channels to appropriate beams (beam selection), which solves a classification problem:

$$\begin{aligned} \mathsf{D_c}\colon \mathbb{R}^d &\longrightarrow \mathbb{N}^* \\ \mathbf{z} &\longrightarrow i = \mathsf{D_c}\left(\mathbf{z}\right). \end{aligned} \tag{8.17}$$

Our intention is to compare the proposed technique with the one presented in [LeMag+22a], where a precoder, not constrained to belong to a codebook, is inferred from the pseudo-location. In that case the decoding function becomes a precoder decoding function, which solves a regression problem:

$$\begin{aligned} \mathsf{D_r}\colon \mathbb{R}^d &\longrightarrow \mathbb{C}^{N_a} \\ \mathbf{z} &\longrightarrow \mathbf{w} = \mathsf{D_r}\left(\mathbf{z}\right). \end{aligned} \tag{8.18}$$

Remark 8.3. *Note that the precoder is learned only for one subcarrier, e.g. the central*

subcarrier: it is possible to learn it for every subcarrier but it is not done in this chapter for the sake of clarity in the comparisons. One can remark that, due to Eq. (8.15)*, the inferred beam by the classification learner is valid for all subcarriers. This also holds true for the regression learner under the narrowband assumption. Note that there is only one encoding function, i.e. at the BS doing the channel charting, but B decoding functions, i.e. one at each BS.*

Performance measures

In order to measure the performance of the beam decoding function, i.e. decoding the correct beam index from the pseudo-location at a particular BS b, it is proposed to use the modified top-k accuracy defined as:

$$\gamma_b^k = \frac{1}{N_u} \sum_{u=1}^{N_u} \mathbb{1}_{\left\{i_{b,u}^{\star}, i_{b,u}^{\star,2}, \cdots, i_{b,u}^{\star,k}\right\}} \left(\mathsf{D}_{\mathsf{c},b}\left(\mathbf{z}_u\right)\right), \tag{8.19}$$

where $\mathsf{D}_{\mathsf{c},b}$ denotes the classification decoder associated with BS b. This metric lies in the $[0, 1]$ range: 0 means that predicted beams are never among the k best, while 1 means that all predicted beams are always among the k best. When learning the precoder, the performance measure at BS b for each UE u is the normalized correlation between the learned precoder and the downlink channel (at the central subcarrier denoted f), as proposed in [LeMag+22b]:

$$\eta_{b,u} = \frac{\left|\mathsf{D}_{\mathsf{r},b}\left(\mathbf{z}_u\right)^{\mathsf{H}} \tilde{\mathbf{g}}_{b,u,f}\right|^2}{\left\|\tilde{\mathbf{g}}_{b,u,f}\right\|_2^2}, \tag{8.20}$$

where $\mathsf{D}_{\mathsf{r},b}$ denotes the regression decoder associated with BS b. This metric is between 0 and 1: 0 means that the precoder is orthogonal to the downlink channel, while 1 means that it is perfectly aligned, i.e. collinear, with the downlink channel.

8.3 Proposed method

This section presents the proposed beam selection method, which can be divided into two distinct stages: inference, representing the normal operating mode, and training, which serves as a calibration phase. Both the encoding and decoding functions follow a similar approach as in [LeMag+22a; Yas+22]. Additionally, as exposed earlier, the encoding stage is done using channel charting only at one BS, referred to as BS1 for the rest of the chapter.

8.3.1 Inference

The inference phase of the proposed method can be split into three distinct steps, which are detailed below and illustrated in Fig. 8.4:

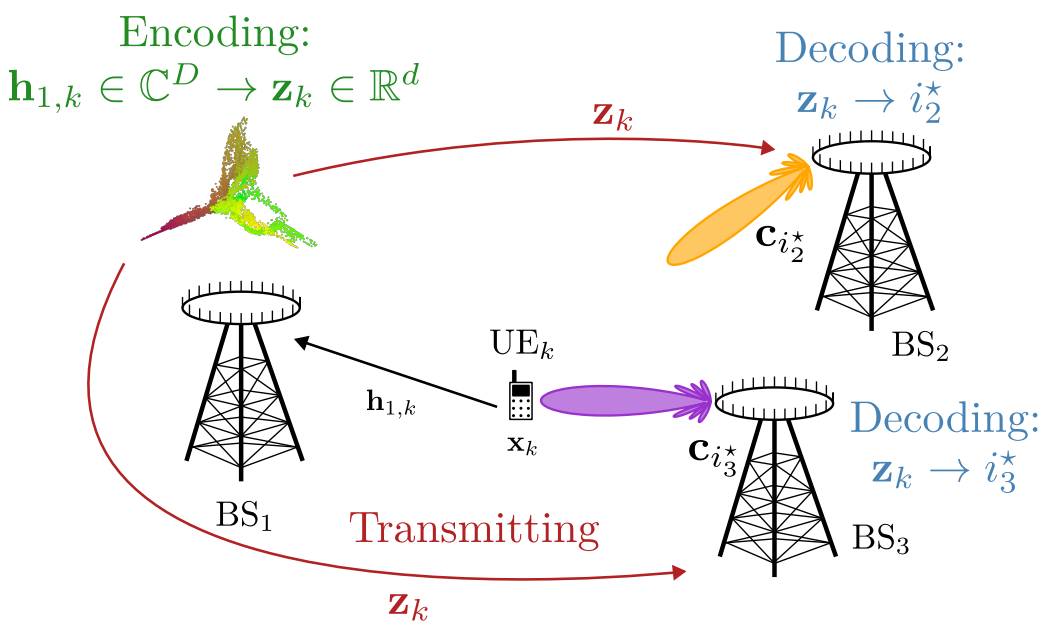


Figure 8.4: Schematic of the proposed encoding/decoding scheme, in the inference state. $\mathbf{c}_i$ denotes the ith codeword of the considered codebook.

- **Encoding.** Using the dimensionality reduction algorithm `ISOMAP` as the channel charting method, d latent variables ($d \ll D$) are computed from an uplink channel $\mathbf{h}_{1,k} \in \mathbb{C}^D$ to form the pseudo-location $\mathbf{z}_k \in \mathbb{C}^d$ at BS1. As it is conventional for manifold learning methods, `ISOMAP` is computationally intensive in out-of-sample scenarios. As this inference method is used for the beam management of each UE, it is of paramount importance to optimize it. A way to overcome this complexity issue has been presented in [LeMag+22a] and is recalled in Fig. 8.5. $\mathbf{D}$ and $\mathbf{Z}$ are matrices composed of uplink calibration channels and associated pseudo-locations, obtained using the `ISOMAP` algorithm equipped with the PI distance recalled in Eq. (8.11). For a given channel $\mathbf{h}_{1,k}$, the associated pseudo-location $\mathbf{z}_k$ can be seen as a wisely chosen convex combination of the pseudo-locations obtained with the calibration uplink channels.

- **Transmitting.** The computed pseudo-location $\mathbf{z}_k$ is sent to all other BSs by BS1.

- **Decoding.** All BSs use the received pseudo-location to perform beam selection using their decoding functions.

A beam management complexity comparison is presented in Table 8.1. The classical approach is beam sweeping: each BS must test each beam of its codebook. For each of the B BSs, this requires the estimation of D complex numbers, representing the uplink channel of a UE. On the other hand, the proposed approach only requires one uplink channel estimation, and the transmission of the computed pseudo-location, of dimension d, to the other BSs, achieving a substantial complexity reduction.

Figure 8.5: Method for efficient channel charting in out-of-sample scenarios [LeMag+22a; Yas+22]. This succession of operations can be seen as an encoding neural network, with parameters $\mathbf{D}$ and $\mathbf{Z}$. The HT_K non-linearity is presented in Section 6.3.

Table 8.1: Inference complexities ($d \ll D$).

	Beam sweeping	Proposed method
Inference complexity	$\mathcal{O}(BD)$	$\mathcal{O}(D + Bd)$

8.3.2 Training

The main objective of this chapter is to propose efficient encoding and decoding functions.

Encoder/decoder architectures

Encoder. As stated before, the encoder consists of the `ISOMAP`-based channel charting method illustrated in Fig. 8.5. This approach enables an efficient out-of-sample charting procedure, and overcomes the well-known limitations of `ISOMAP` in such contexts. It relies on two matrices, $\mathbf{D}$ and $\mathbf{Z}$, that are obtained using `ISOMAP`. Specifically, an uplink channel calibration dataset $\{\mathbf{h}_{1,1}, \cdots, \mathbf{h}_{1,N_{\text{cal}}}\}$ is first collected at BS1. The `ISOMAP` algorithm is then applied using the PI distance proposed in [LeMag21a], resulting in a pseudo-location calibration dataset $\{\mathbf{z}_1, \cdots, \mathbf{z}_{N_{\text{cal}}}\}$. The matrices are subsequently defined as $\mathbf{D} = [\mathbf{h}_{1,1}, \cdots, \mathbf{h}_{1,N_{\text{cal}}}] \in \mathbb{C}^{D \times N_{\text{cal}}}$ and $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_{N_{\text{cal}}}] \in \mathbb{C}^{d \times N_{\text{cal}}}$.
Decoder. It is proposed to use a neural network to learn the decoding function at each BS: the input is a pseudo-location and the output is a probability vector representing the probability of selection for each beam in the codebook. Domain knowledge can be incorporated into the design of the decoding function. Specifically, one can notice that there exist regions, e.g. close to the BS, in which the optimal beam varies rapidly with the location, indicating that the spatial best-beam distribution contains high-frequency components. To account for this, a RFF neural architecture [RR07; Tan+20] can be used. Indeed, as presented in Section 4.1.1, classical MLP networks are biased towards learning low frequency functions [Rah+19; Cao+21], and RFF architectures were proposed to overcome this limitation. The proposed decoding network is illustrated in Fig. 8.6, where the output of the RFF block expresses as:

$$\mathbf{f} = \begin{bmatrix} \cos(2\pi \mathbf{B}\mathbf{z}) \\ \sin(2\pi \mathbf{B}\mathbf{z}) \end{bmatrix}, \tag{8.21}$$

with $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_d] \in \mathbb{R}^{F \times d}$ initialized as $\mathbf{b}_i \sim \mathcal{N}(\mathbf{0}_{\mathbb{R}^F}, \sigma^2 \mathbf{Id}_F)$, where σ is a hyperparameter controlling the frequency range of the RFF embedding layer. The higher the

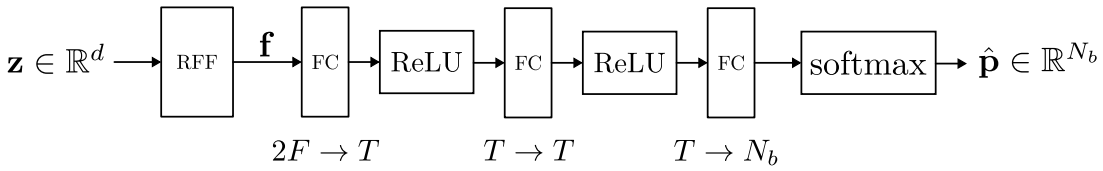


Figure 8.6: Proposed decoder neural architecture: RFF network based on [LeMag+22a].

σ, the easier it will be for the network to learn high frequency content. Note that $\mathbf{B}$ is learned during training.

Remark 8.4. *Note that, in contrast to Eq.* (8.17)*, the output of the decoding function, presented in Fig. 8.6, is not a scalar as it lies in* $\mathbb{R}^{N_b}$*: it does not correspond to the hard decisions typical of a classification problem. This is simply an implementation detail: since this output represents the probability of each codeword given the input pseudo-location, computing the argmax of this probability vector yields the best beam index.*

During training, the loss function to minimize at BS b is the multiclass cross-entropy defined as:

$$\mu_b = -\sum_{(\mathbf{z},\mathbf{p}_b)\in\mathcal{D}} \mathbf{p}_b^{\mathsf{T}} \log_2 \mathsf{D}_{\mathsf{c},b}(\mathbf{z}), \tag{8.22}$$

where $\mathcal{D} = \{(\mathbf{z}_u, \mathbf{p}_{b,u})\}_{u=1}^{N_t}$ is the training set, and $\mathbf{p}_{b,u} \in [0,1]^{N_b}$ is the true beam probability vector for UE u at BS b, i.e.

$$(\mathbf{p}_{b,u})_l = 1 \Leftrightarrow i^\star_{b,u} = l. \tag{8.23}$$

Remark 8.5. *It should be noted that domain knowledge, presented in (H1), was leveraged for the design of both the encoding and decoding functions. Specifically, as discussed in Section 8.1, the motivation for using the PI distance originates from (H1). This PI distance is incorporated into the encoding function presented in Fig. 8.5, where the matrices* $\mathbf{D}$ *and* $\mathbf{Z}$ *are initialized using the* `ISOMAP` *algorithm equipped with the PI distance. Furthermore, the decoding function relies on the assumption of rapid variations in the spatial best-beam distribution, which also traces back to (H1). As shown in Chapter 4, the fast variations of the location-to-channel mapping can be explained from (H1), which in turn induce fast variations in the optimal beams, since precoders and channels are inherently linked.*

8.4 Experiments

8.4.1 Learning framework and baselines

Simulation settings

The experiments are carried out on the same dataset as in [LeMag+22a]. Namely, the `DeepMIMO` [Alk19] urban outdoor "O1" dataset is used, with $B = 2$ BSs equipped with

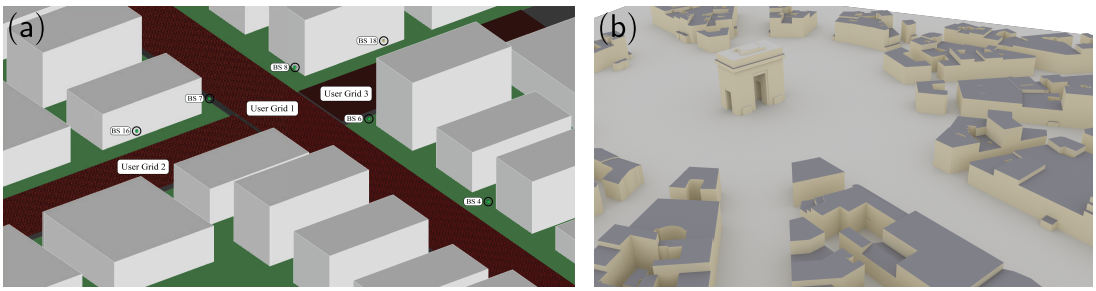


Figure 8.7: 3D view of the considered scenes: (a): `DeepMIMO`, (b): `Sionna`.

UPAs with $N_a = 64$ antennas. $N_s = 16$ subcarriers are considered over a 20 MHz bandwidth in both the uplink and downlink. Another dataset is generated with the same parameters, using the `Sionna` [Hoy+22] ray-tracing library. This dataset is generated in the *Etoile* scene in Paris, France. Both scenes are illustrated in Fig. 8.7 For both scenarios, the uplink central frequency f_{ul} is 3.5 GHz, while the downlink central frequency f_{dl} is 28 GHz. In the `DeepMIMO` scene, there are $N_u = 4.2$k training UEs (UE spatial density: 0.17 UEs/m^2), while there are $N_u = 7$k training UEs in the `Sionna` scene (UE spatial density: 0.042 UEs/m^2). One can remark that the training UE spatial density is quite low, making it similar to what could be encountered in actual systems.

Network parameters and baselines

For the proposed architecture, $F = 200$ frequencies are considered, and the hidden layer size is $T = 64$. The baseline for comparison is a MLP neural network, with the same number of parameters as the proposed architecture: it is the same network as in Fig. 8.6, but the input layer is replaced by a classical fully connected layer. Another baseline is the nearest-neighbor (1-NN) ML method: for a given test pseudo-location, the 1-NN predicted beam will be the best beam of the nearest train pseudo-location.

8.4.2 Results

Best beam spatial distribution

In Fig. 8.8, the best beam spatial distribution for the `DeepMIMO` and `Sionna` datasets is depicted. Each precoder in the codebook ($N_b = 256$) is color-coded with a unique color. It can be observed that the best beam changes very quickly near BS2 (red triangle) in both the `DeepMIMO` and `Sionna` datasets.

Charting performance

The channel charting map can be seen in Fig. 8.9. One can remark that UEs close to BS1 (UEs in purple), appear more spaced-out in the pseudo-location space: this phenomenon is due to the fact that the angular resolution of antenna arrays is independent of the

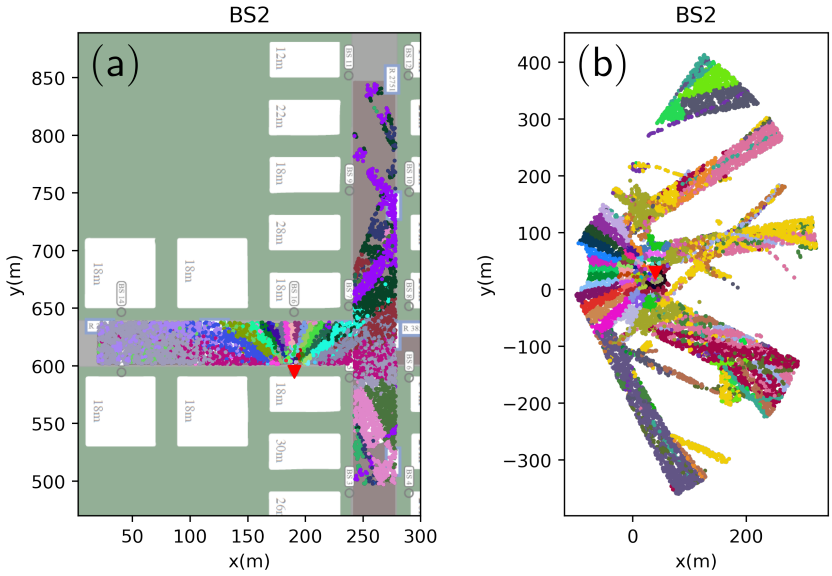


Figure 8.8: Best beam spatial distribution: (a): `DeepMIMO`, (b): `Sionna`. The red triangle denotes the location of BS2.

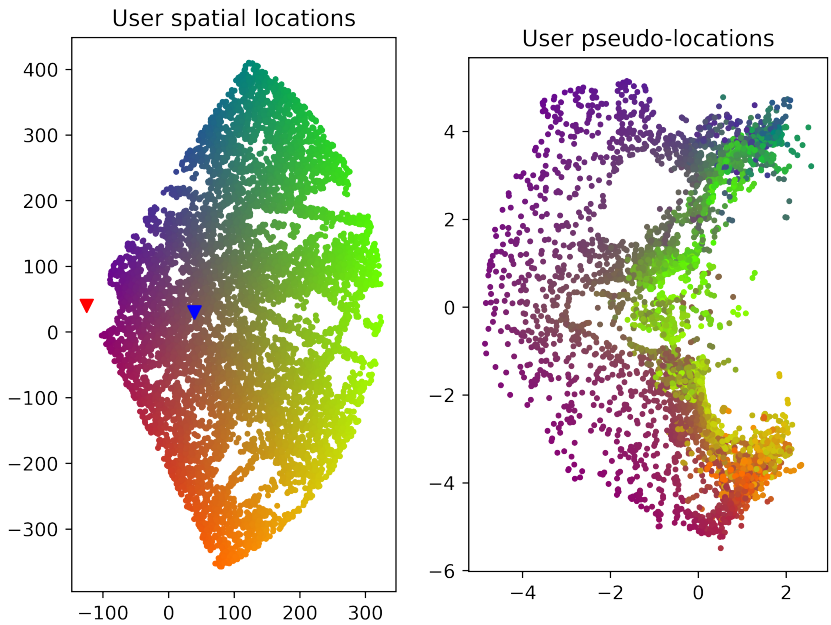


Figure 8.9: Channel charting map (`Sionna`). The red and blue triangles respectively denotes the location of BS1 and BS2.

Table 8.2: Charting metrics (5% neighborhoods).

	TW (↑)	CT (↑)	KS (↓)
`DeepMIMO`	0.973	0.929	0.471
`Sionna`	0.960	0.952	0.292

Table 8.3: Beam prediction accuracy.

`DeepMIMO`	Proposed (RFF)	MLP	1-NN
Top 1 acc. $= \gamma_2^1$ (%)	**66.07**	56.06	61.40
Top 3 acc. $= \gamma_2^3$ (%)	**90.66**	85.09	88.77
`Sionna`	Proposed (RFF)	MLP	1-NN
Top 1 acc. $= \gamma_2^1$ (%)	66.07	54.07	**69.73**
Top 3 acc. $= \gamma_2^3$ (%)	78.27	69.07	**81.87**

distance, whereas the tangential displacement corresponding to a given angular deviation is proportional to the distance [2]. Channel charting metrics for both datasets are presented in Table 8.2. As presented in Section 8.1, all CC-related metrics, that is TW, CT and KS, are within the $[0, 1]$ range. Both TW and CT are optimal at 1 while KS is optimal at 0. One can remark in Table 8.2 that, for both datasets, TW and CT are very good, while KS is good for the `Sionna` dataset and average for the `DeepMIMO` dataset.

Beam prediction performance

The beam prediction performance at BS2, i.e. the one not used for channel charting can be found in Table 8.3. Note that the proposed RFF network consistently outperforms the MLP baseline. It is worth noting that the 1-NN approach works well in the `Sionna` dataset, with slightly better performance than the RFF network. It is interesting, as this shows that the information contained in the low dimensional pseudo-location is sufficient for a very simple ML method, such as the 1-NN, to have good performance. One can remark that, even if the top-1 accuracy performance remains modest, the top-3 performance is good in both datasets. This means that using the pseudo-locations allows to have good performance in predicting one of the three best beams in the codebook. This does not result in a huge performance loss due to the high beam density in a 256 beams 2D-DFT codebook: the second and third best beams still have high correlations.

2. Note that ways to quantify this precisely are detailed in [Yas+24].

Table 8.4: Inference time.

	RFF	MLP	1-NN (a)	1-NN (b)
Exec. time [ns]	602.6	**145.8**	4928.2	10913.9

Inference time

It is proposed to compare the inference time of the different approaches in Table 8.4: 1-NN (a) refers to the optimized ball-tree 1-NN, while 1-NN (b) refers to the brute force 1-NN. One can see that the neural network approaches have faster inference times: this is can be explained by their GPU implementation. Note that a GPU-optimized 1-NN could be of particular interest, as Table 8.3 shows that this method performs well with pseudo-locations. However, one has to keep in mind that, when considering online learning, the neural network approaches would outperform the 1-NN in inference complexity, as for each new sample, the 1-NN approach would have to compute an entire distance matrix whose size would grow with time.

Precoder learning performance

In this experiment, the goal is to learn the precoder from the pseudo-location. In contrast to the previous experiment, which involved a classification task, this one corresponds to a regression task. A neural network, referred to as RFF (regr.) is defined as the same network than in Fig. 8.6, only dropping the last softmax non-linearity and adapting the last fully connected layer size so that the output is a precoding vector $\mathbf{w} \in \mathbb{C}^{N_a}$. The same holds true for the MLP baseline, denoted MLP (regr.). The same correlation-based training as in [LeMag+22b] is used. The 1-NN baseline is defined as follows: for a given test pseudo-location, the associated precoder is the normalized downlink channel of the nearest training pseudo-location. The performance of the beam prediction networks, i.e. RFF (classif.) and MLP (classif.), are also presented for this regression task: for a given pseudo-location each network infer a precoder in the codebook, and then its correlation with the downlink channel at the central frequency is computed, as in Eq. (8.20). The performance metric is the cumulative distribution function (CDF) of the normalized correlation between the learned precoder and the downlink channel at the central frequency. In Fig. 8.10, it is shown that the regression networks present better performance than the classification networks, for both datasets. This stems from the absence of constraints on the azimuth and elevation beam steering angles, as well as on the lack of requirement for the learned precoder to be a pure steering vector. This flexibility allows the regression network to learn the optimal precoder directly, unlike classification networks, where angular constraints are imposed by the codebook. One can also note that, for the `DeepMIMO` dataset, RFF and MLP regression networks outperform the 1-NN approach. The performance gain of the regression networks is illustrated in Fig. 8.11: regression networks offer a higher correlation coverage than classification networks.

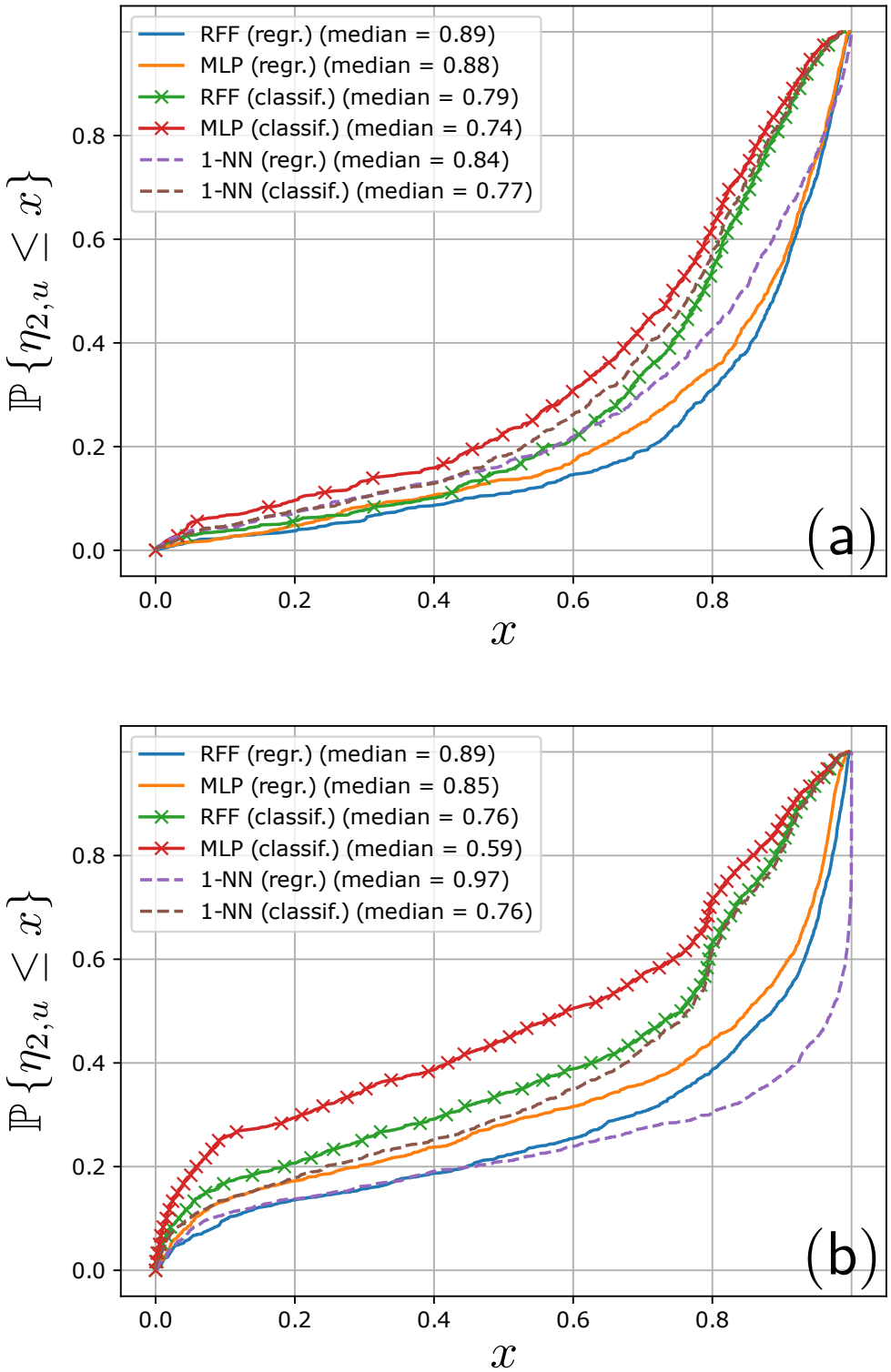


Figure 8.10: CDF of the correlations at BS2, (a): `DeepMIMO`, (b): `Sionna`.

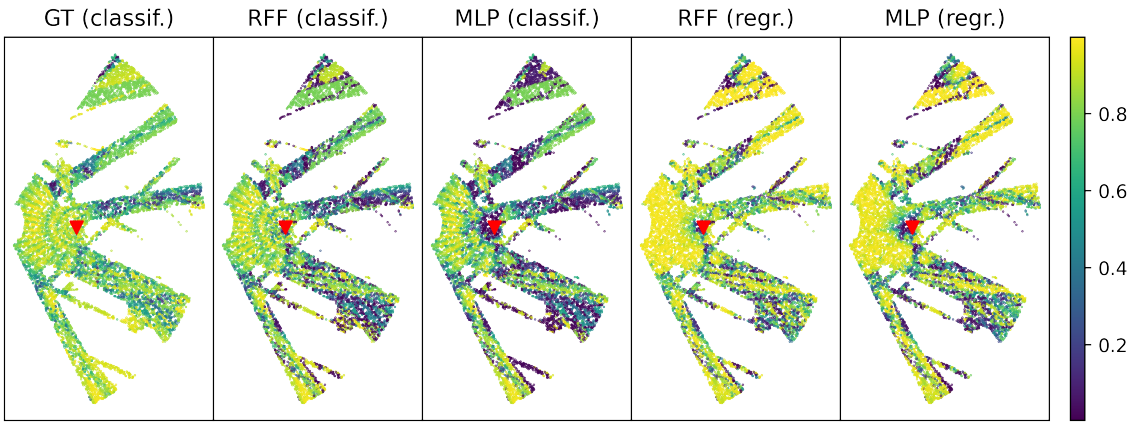


Figure 8.11: Normalized correlation maps, $\eta_{2,u}$ at BS2, `Sionna`.

8.5 Conclusion

This chapter considered a joint encoder/decoder design problem for a CSI compression scheme in a cell-free mMIMO system, as formalized in Problem 3.4. The main takeaways are summarized as follows.

- The MB-ML paradigm has been successfully applied to this problem. Specifically, both the encoding and decoding functions consist of neural networks whose architectures are **structured** based on domain knowledge formalized in (H1). The encoder leverages channel charting for efficient dimensionality reduction, while the decoder relies on a RFF-based architecture to accurately map encoded channels to best-beam codebook indices. Additionally, the encoder is **initialized** based on domain knowledge, as its structure, based on (H1), constrains the encoder parameters to represent a set of channels and the corresponding embedded pseudo-locations.

- Pseudo-locations, i.e., compressed channels obtained via channel charting, were shown to enable beam selection at BSs other than the one performing channel charting. This yields an efficient beam allocation scheme, as CSI estimation is required at only a single BS in the entire cell-free system.

- Evaluation in realistic propagation environments showed that the proposed approach outperforms both classical and neural baselines. Moreover, low-dimensional pseudo-locations were shown to be sufficiently descriptive for simple baseline methods, such as 1-NN, to achieve strong performance.

- Finally, it was shown that directly learning a precoder, rather than learning a codeword in a codebook, improves overall performance.

Chapter 9

Channel charting and CSI compression

This chapter focuses on designing an efficient CSI compression method for FDD schemes in MU-MIMO systems. Since CSI is primarily used for beamforming, the proposed method bypasses CSI reconstruction and directly infers a precoder. Specifically, the channels are compressed into low-dimensional pseudo-locations via channel charting at each UE, which are then transmitted to the BS and decoded into a precoder. Section 9.1 presents a general introduction to the CSI compression problem. Section 9.2 presents in details the proposed approach. Section 9.3 introduces experimental results that validate the proposed method, and Section 9.4 summarizes the contributions of this chapter.

Contents

9.1 Problem setting **226**
- 9.1.1 General introduction 226
- 9.1.2 Contributions and related work 226
- 9.1.3 Problem formulation 227

9.2 Proposed approach **230**
- 9.2.1 Encoder/decoder architectures 230
- 9.2.2 Learning strategy 232
- 9.2.3 Parameter optimization 232

9.3 Experiments **233**
- 9.3.1 Simulation environment and baselines 233
- 9.3.2 Results 234

9.4 Conclusion **238**

This chapter is mainly based on a conference article presented at the *2024 58th Asilomar Conference on Signals, Systems, and Computers* [Cha+24a]. Furthermore, the experiments described in this chapter entailed a total computational cost of approximately 1 GPU day.

9.1 Problem setting

9.1.1 General introduction

The shift towards higher operating frequencies in 5G and B5G communication systems has led to the widespread use of antenna arrays made up of a large number of smaller radiating elements, capable of forming high-gain directional beams. However, this increase in the BSs antenna number causes a significant overhead in the CSI reporting procedures used in FDD schemes. Indeed, in FDD systems, in order to perform downlink channel estimation, UEs need to periodically report one complex number per antenna and subcarrier to the BS. To mitigate this reporting overhead, CSI compression techniques have been envisioned [WSJ18]. The associated reporting procedure can be summarized as follows: in a FDD system, a BS sends pilot symbols to a UE that uses them to perform CSI estimation. The UE then compresses the estimated CSI and transmits it to the BS. Leveraging its readily available computing power, the BS is then able to reconstruct the CSI with minimal distortion, before using it to perform communication tasks such as beamforming or resource allocation. This problem has been extensively studied in the past few years, with AI/ML methods proposed to address it. A non-exhaustive list of relevant studies includes [WSJ18; Guo+20; Wan+19; LWS20; Che+22; CGS22]. An in-depth review of the deep learning based CSI compression literature is provided in [Guo+22].

This chapter considers the CSI compression problem from a task-oriented perspective, and studies the use of channel charting in this context. Since the reconstructed CSI is primarily used for beamforming, one could consider the CSI compression problem where the decoding step reconstructs a precoding matrix rather than the CSI. As presented in Section 8.1, CC [Stu+18] is an unsupervised dimensionality reduction method that compresses CSI into chart locations, with the objective of preserving spatial neighborhoods. CC has been used for beam prediction [Pon+21; Yas+23], precoder learning [LeMag+22a], but also for positioning [ESB23; TPS25]. In the context of CSI compression, CC clearly appears as a promising candidate for the encoding stage, as, by definition, a chart location is a compressed version of its associated CSI.

9.1.2 Contributions and related work

Contributions

This chapter addresses the concept of task-based CSI compression, which aims to reduce CSI reporting overhead while directly optimizing beamforming performance. The main contributions are summarized as follows.

- A neural encoder and decoder architectures are introduced, both structured based on domain knowledge, formalized in (H1). The encoder maps CSI to low-dimensional pseudo-locations via channel charting, while the decoder maps these to a precoding matrix.

- An end-to-end heuristic training strategy is proposed, ensuring beamforming performance upon convergence.
- The number of learnable parameters is reduced through a physically grounded subsampling strategy [Tan+23] applied to the encoder, yielding a ninetyfold reduction with negligible performance loss.
- Simulations on realistic data show that the proposed approach achieves excellent beamforming performance with significantly higher compression ratios (around 1024) compared to those obtained by classical auto-encoders (around 64).

Related work

Inferring a precoder from a chart location has been studied for cell-free massive MIMO systems in [LeMag+22a]. However, that study did not optimize the encoder complexity and only trained the decoder network. Similar studies have been carried out with precoders constrained in a codebook in [Pon+21; Yas+23]. Jointly optimizing CSI compression and beamforming performance has been studied in [GWJ21; Sun+22; Xue+23]. However, none of these studies considered channel charting for the encoding stage. Finally, CSI compression using AI/ML methods have been extensively studied in [WSJ18; Guo+20; Wan+19; LWS20; Che+22; CGS22]. These approaches obtain good performance in CSI reconstruction but often present poor compression ratios compared to our approach, as they are not specialized to a given task.

9.1.3 Problem formulation

This chapter considers a MU-MIMO system comprised of a BS equipped with N_a antennas and N_u mono-antenna UEs. The BS transmits symbols to the UEs on N_s subcarriers.

Reporting procedure

Fig. 9.1 depicts the system model as well as the proposed reporting procedure, which can be detailed as follows:

1. A BS sends pilot symbols to a set of UEs.
2. The UEs use the pilots to estimate their downlink channel $\mathbf{h}_k \in \mathbb{C}^D$. Note that vectorized channels are considered, so that $D = N_a N_s$.
3. Then, UEs perform CSI compression: the downlink channel is encoded into a low dimensional vector $\mathbf{z}_k \in \mathbb{R}^d$.
4. Finally, the BS receives the encoded channels from all UEs and decode them into a precoding matrix $\mathbf{W}$, used to perform spatial multiplexing.

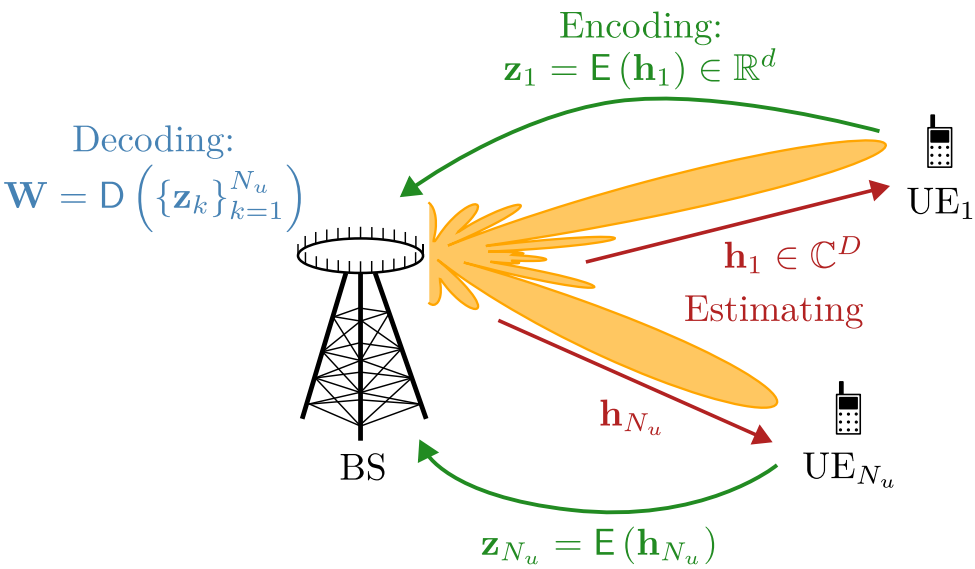


Figure 9.1: System model, reporting and encoding/decoding procedures.

Objective

The goal of this chapter is to design an efficient CSI compression method that solves the following sum-rate (SR) optimization problem in a parallel transmission to $K \leq N_u$ UEs:

$$\begin{aligned} \underset{\mathbf{W} \in \mathbb{C}^{D \times K}}{\text{maximize}} \quad & \sum_{k=1}^{K} \log_2 \left(1 + \frac{\left|\mathbf{h}_k^{\mathsf{H}} \mathbf{w}_k\right|^2}{\sigma_k^2 + \sum\limits_{j \neq k} \left|\mathbf{h}_k^{\mathsf{H}} \mathbf{w}_j\right|^2} \right), \\ \text{subject to} \quad & \mathsf{C}_p : \forall k \in [\![1, K]\!], \|\mathbf{w}_k\|_2 = \frac{1}{\sqrt{K}} \end{aligned} \tag{9.1}$$

where $\mathbf{w}_k$ is a precoding vector decoded from a compressed representation of the channel $\mathbf{h}_k$ and σ_k^2 is the associated noise variance. The power constraint C_p considers uniform power allocation between each UE.

The foundation of the proposed method is formally based on defining a well suited encoding function E and decoding function D. The role of the encoding function is to project the CSI into a low-dimensional space:

$$\begin{aligned} \mathsf{E} \colon \mathbb{C}^D &\longrightarrow \mathbb{R}^d \\ \mathbf{h} &\longrightarrow \mathbf{z} = \mathsf{E}(\mathbf{h}), \end{aligned} \tag{9.2}$$

where $d \ll D$. Conversely, the decoding function maps the low-dimensional encoded

channels to a precoding matrix:

$$\begin{aligned} \mathsf{D}\colon \mathbb{R}^d \times \cdots \times \mathbb{R}^d &\longrightarrow \mathbb{C}^{D\times K} \\ \{\mathbf{z}_k\}_{k=1}^K &\longrightarrow \mathbf{W} = \mathsf{D}\left(\{\mathbf{z}_k\}_{k=1}^K\right). \end{aligned} \tag{9.3}$$

Remark 9.1. *The optimization problem presented in Eq.* (9.1)*, together with the presented encoding and decoding function, is a direct expression of Problem 3.4. Indeed, the objective is to optimize a communication metric, namely the sum-rate, by designing an efficient encoding/decoding framework. It should be noted that, unlike the formulation in Problem 3.4, Eq.* (9.1) *represents a maximization problem rather than a minimization problem. This distinction is minor, since maximizing an objective function is equivalent to minimizing its negative counterpart.*

Performance measure

The performance of the proposed method is measured in different configurations: mono and multi-UE scenarios. In a mono-UE scenario, it has been shown in [LeMag+22a] that the transmission capacity is linked to the squared cosine similarity between the channel $\mathbf{h}_k$ and the associated inferred precoder $\mathbf{v}_k$:

$$\rho_k = \frac{\left|\mathbf{v}_k^{\mathsf{H}}\mathbf{h}_k\right|^2}{\|\mathbf{v}_k\|_2^2\,\|\mathbf{h}_k\|_2^2}. \tag{9.4}$$

This metric is normalized within the $[0,1]$ range. 0 means that the precoder is orthogonal to the channel, leading to no transmission, while 1 means that the precoder is perfectly collinear with the channel. The mean and median similarities over the UE set will be considered.

In a multi-UE scenario, the performance measure is the sum-rate averaged across distinct groups of UEs in the location space. Considering the partition of the UE set $\mathcal{T}_u$ into B groups of K UEs:

$$\mathcal{T}_u = \bigcup_{b=1}^{B} \mathcal{T}_{u,b}, \tag{9.5}$$

with $|\mathcal{T}_{u,b}| = K$ and $\forall m \neq n, \mathcal{T}_{u,m} \cap \mathcal{T}_{u,n} = \varnothing$, the sum-rate can be defined as:

$$\mathsf{SR} = \frac{1}{B}\sum_{b=1}^{B}\sum_{k}\log_2\left(1 + \frac{\left|\mathbf{h}_{k,b}^{\mathsf{H}}\mathbf{w}_{k,b}\right|^2}{\sigma_{k,b}^2 + \sum\limits_{j\neq k}\left|\mathbf{h}_{k,b}^{\mathsf{H}}\mathbf{w}_{j,b}\right|^2}\right), \tag{9.6}$$

where $\mathbf{h}_{i,b} \in \mathcal{T}_{u,b}$ is the channel for the ith UE of the bth group, and $\mathbf{w}_{i,b}$ is the associated decoded precoder.

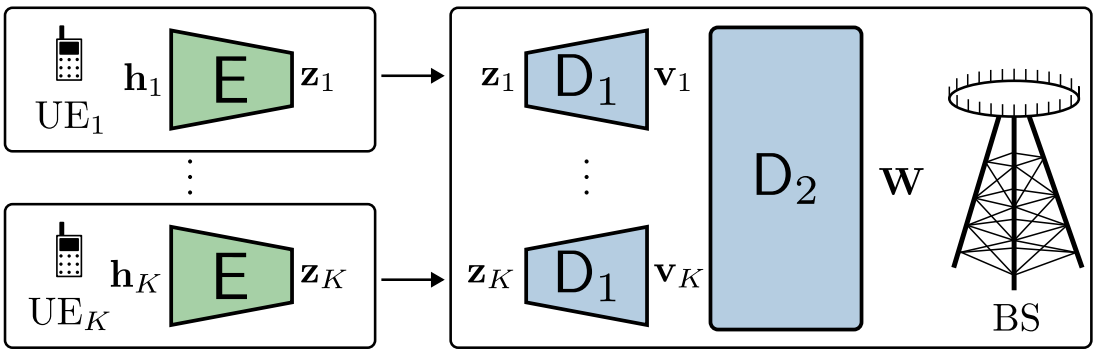


Figure 9.2: Overview of the proposed approach: encoder (left), decoder (right).

9.2 Proposed approach

This section presents the chosen models for the encoding and decoding functions, as well as the learning strategy and the learnable parameter reduction method. The CC-based encoder and decoder models are similar to those proposed in [LeMag+22a; Yas+22; Yas+23]. An overview of the proposed approach is presented in Fig 9.2.

9.2.1 Encoder/decoder architectures

Encoder

As in [LeMag+22a; Yas+22; Yas+23], the encoding function is done through CC with a neural model presented in Fig. 9.3. This model is initialized through matrices $\mathbf{D}$ and $\mathbf{Z}$. The matrix $\mathbf{D} = [\mathbf{h}_1, \cdots, \mathbf{h}_{N_c}] \in \mathbb{C}^{D \times N_c}$ consists of the concatenation of N_c calibration channel vectors, while $\mathbf{Z}$ is initialized through the `ISOMAP` algorithm. This algorithm embeds the calibration channels into chart locations $\mathbf{z}_k \in \mathbb{R}^d$ ($d \ll D$). Note that the phase-insensitive distance of [LeMag21a] is used in the `ISOMAP` algorithm [1]. The chart locations are then concatenated into the matrix $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_{N_c}] \in \mathbb{R}^{d \times N_c}$. As many other dimensionality reduction methods, `ISOMAP` is known to be computationally intensive in out-of-sample scenarios, which reduces the efficiency of the encoding stage for channels not included in the calibration set. The presented CC model aims to address this issue by mapping an unseen channel $\mathbf{h}_j$ into a wisely chosen convex combination of calibration chart locations [2].

Decoder

It is proposed to split the decoder into two parts, namely D_1 and D_2, so that $\mathsf{D} = \mathsf{D}_2 \circ \mathsf{D}_1$, as presented in Fig. 9.3. The first stage, D_1, allows to map the chart locations to mono-UE precoding vectors. Then, the second stage, D_2, is designed to concatenate and process the

1. A more detailed presentation of this distance and its motivation is provided in Section 8.1.
2. Note that the HT_K non-linearity is presented in Section 6.3.

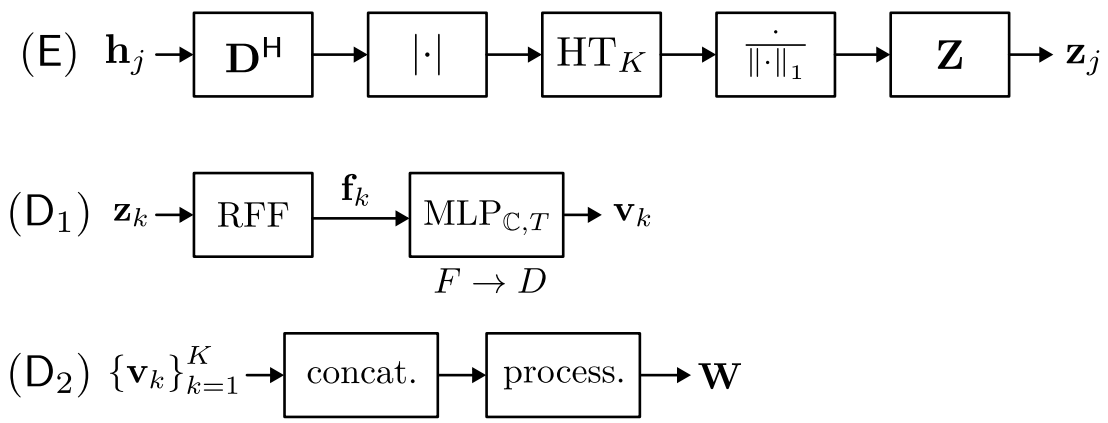


Figure 9.3: Encoder, based on [LeMag+22a; Yas+22], and decoder architectures.

mono-UE precoding vectors into a multi-UE precoding matrix. The mapping of chart locations to precoding vectors is done through a neural network. Note that this mapping can present relatively fast variations in the embedding space. Indeed, the optimal precoders may vary rapidly in some regions of the location space. As channels originating from nearby locations are expected to be embedded to close chart locations through CC, this fast variation behavior is expected to be translated in the embedding space. As a result, it is proposed to use a RFF [RR07] network for the chart-location-to-precoder mapping learning. Indeed, since classical neural architectures such as MLPs have been shown to be biased towards learning low frequency functions [Rah+19; Cao+21; JohXu+20], RFF networks have been proposed as a solution to this spectral bias issue [Tan+20] [3]. The RFF block is defined as:

$$\mathbf{f}_k = \begin{bmatrix} \cos\left(2\pi \mathbf{B}\mathbf{z}_k\right) \\ \sin\left(2\pi \mathbf{B}\mathbf{z}_k\right) \end{bmatrix}, \tag{9.7}$$

where $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_d] \in \mathbb{C}^{F \times d}$ is a frequency matrix initialized as $\mathbf{b}_i \sim \mathcal{N}\left(\mathbf{0}_{\mathbb{R}^F}, \sigma_B^2 \mathbf{Id}_F\right)$. Note that σ_B is an hyperparameter linked to the variation intensity in the chart location space: a higher σ_B facilitates the decoder network's ability to learn rapid variations in the target mapping. The $\mathrm{MLP}_{\mathbb{C},T}$ block denotes a three-layer MLP of size T, with $\mathtt{ReLU}_{\mathbb{C}}$[4] activation functions, and complex weights and biases.

The second decoding stage consists of a concatenation and a processing block. The role of the processing block is to introduce inter-UE interference cancellation in the concatenated precoding matrix. This could be achieved using a neural network; however, the proposed approach leverages classical linear precoding techniques such as MRT or MMSE. These methods are grounded in well-established statistical criteria, offering robustness and interpretability. Thus, the processing block of D_2 only consists of a denormalization step, linear matrix operations, and a normalization step to respect the uniform power allocation constraint. A more comprehensive discussion on this matter is presented below.

3. Detailed information about the spectral bias are provided in Section 4.1.1
4. Note that $\forall z_1 \in \mathbb{C}, \mathtt{ReLU}_{\mathbb{C}}\left(z_1\right) = \mathtt{ReLU}\left(\mathfrak{Re}\left\{z_1\right\}\right) + \mathtt{jReLU}\left(\mathfrak{Im}\left\{z_1\right\}\right)$.

Remark 9.2. *It should be noted that both the proposed encoder and decoder architectures build on the domain knowledge presented in (H1). Since the proposed encoder/decoder architectures are based on [Yas+23], which is presented in details in Chapter 8, the reader is referred to the detailed explanation in Remark 8.5.*

9.2.2 Learning strategy

It is proposed to train the proposed model using an end-to-end approach and a task-oriented loss function. As the objective is to maximize the sum-rate, one could use the negative sum-rate of Eq. (9.6) as a loss function. However, Eq. (9.6) depends on the UEs noise variance and on the number of parallel UEs. This implies that, when using Eq. (9.6) as a loss function, the trained encoder and decoder would be optimal only for a fixed noise variance and a fixed number of parallel UEs, limiting the practical interest of the proposed method. It is suggested to use a heuristic learning strategy that considers the following loss function:

$$\mu = 1 - \frac{1}{|\mathcal{D}|} \sum_{\mathbf{h} \in \mathcal{D}} \frac{\left|\mathbf{v}^{\mathsf{H}} \mathbf{h}\right|^2}{\|\mathbf{h}\|_2^2}, \tag{9.8}$$

where $\mathcal{D}$ denotes the training channel set and $\mathbf{v} = (\mathsf{D}_1 \circ \mathsf{E})(\mathbf{h})$. One can show that Eq. (9.8) is minimized when the precoders are set to $\mathbf{v}^\star = \mathbf{h}\mathrm{e}^{\mathrm{j}\phi} / \|\mathbf{h}\|_2$. Thus, under the constraint of effective loss convergence, the trained decoder D_1 provides the normalized channel affected by an arbitrary phase shift. Through denormalization, one can then access to a close approximation of the channel $\mathbf{h}$. Such channel can then be concatenated into a matrix, and used to form a precoding matrix, following the classical formulas of linear precoders. Note that, when considering this heuristic approach, the final decoding stage D_2 does not present any learnable parameters so that the learning is done on the parameters of the D_1 decoding stage.

9.2.3 Parameter optimization

The learnable parameters are conveyed by matrices $\mathbf{D} \in \mathbb{C}^{D \times N_c}$ and $\mathbf{Z} \in \mathbb{C}^{d \times N_c}$, frequency matrix $\mathbf{B} \in \mathbb{C}^{F \times d}$, in addition to the weights and biases of the decoder's MLP. As $d \ll D$, the main parameter complexity comes from the encoder. Indeed, N_c needs to be high in order to have a good initialization in the `ISOMAP` algorithm, resulting in a large matrix $\mathbf{D}$. Optimizing the encoder parameter complexity is of paramount importance since the encoding stage is done at the UE level, but trained at the BS. Once trained, the BS needs to send the learned encoder parameters to the UEs: reducing the number of parameters in the encoder is then equivalent to reducing the encoder update overhead.

In order to reduce this parameter complexity, it is proposed to introduce an intelligent subsampling of matrices $\mathbf{D}$ and $\mathbf{Z}$. Firstly, $\mathbf{D}$ and $\mathbf{Z}$ are initialized using `ISOMAP` as aforementioned. Then, the similarity subsampling algorithm proposed in [Tan+23], and recalled in Algorithm 7, is applied. Its objective is to discard channels that produce chart

Algorithm 7 Similarity subsampling [Tan+23].

Inputs: Let $\tilde{\mathbf{D}} \in \mathbb{C}^{D\times\tilde{N}}$, $\tilde{\mathbf{Z}} \in \mathbb{R}^{d\times\tilde{N}}$, $\mathbf{D}^c \in \mathbb{C}^{D\times\left(N_c-\tilde{N}\right)}$, $\mathbf{Z}^c \in \mathbb{R}^{d\times\left(N_c-\tilde{N}\right)}$, s.t. $\mathbf{D} = \tilde{\mathbf{D}} \cup \mathbf{D}^c$, $\mathbf{Z} = \tilde{\mathbf{Z}} \cup \mathbf{Z}^c$.

1: **for each** $\mathbf{h}_i \in \mathbf{D}^c$ **do**
2: $\quad s \leftarrow \max\limits_{\mathbf{h}_j\in\tilde{\mathbf{D}}} \operatorname{sim}(\mathbf{h}_i, \mathbf{h}_j)$
3: $\quad \{k, l\} \leftarrow \underset{\substack{1\leq m\neq n\leq\tilde{N} \\ (\mathbf{h}_m,\mathbf{h}_n)\in\tilde{\mathbf{D}}}}{\arg\max} \operatorname{sim}(\mathbf{h}_m, \mathbf{h}_n)$
4: $\quad$ **if** $s < \operatorname{sim}(\mathbf{h}_k, \mathbf{h}_l)$ **then**
5: $\quad\quad$ With probability $p : r \leftarrow k$, otherwise $r \leftarrow l$
6: $\quad\quad \tilde{\mathbf{D}} \leftarrow \left(\tilde{\mathbf{D}} \setminus \mathbf{h}_r\right) \cup \mathbf{h}_i$
7: $\quad\quad \tilde{\mathbf{Z}} \leftarrow \left(\tilde{\mathbf{Z}} \setminus \mathbf{z}_r\right) \cup \mathbf{z}_i$
8: $\quad$ **end if**
9: **end for**

Output: $\tilde{\mathbf{D}}$ (Subsampled channels), $\tilde{\mathbf{Z}}$ (Subsampled chart locations)

locations that are too similar. Indeed, similar chart locations can be considered as redundant, thereby providing no additional useful information. Theoretically, removing this redundancy from the calibration dataset allows to reduce the number of learnable parameters without impacting the global performance. From a formal perspective, Algorithm 7 aims to solve the following minimax problem:

$$\min \max_{(\mathbf{h}_i,\mathbf{h}_j)\in\tilde{\mathbf{D}}} \operatorname{sim}(\mathbf{h}_i, \mathbf{h}_j), \tag{9.9}$$

where $\tilde{\mathbf{D}}$ is the subsampled channel matrix and $\operatorname{sim}(\mathbf{h}_i, \mathbf{h}_j)$ is a similarity metric defined as the cosine similarity of the embedded channels:

$$\operatorname{sim}(\mathbf{h}_i, \mathbf{h}_j) = \frac{\left|\mathsf{E}(\mathbf{h}_i)^{\mathsf{H}} \mathsf{E}(\mathbf{h}_j)\right|}{\|\mathsf{E}(\mathbf{h}_i)\|_2 \|\mathsf{E}(\mathbf{h}_j)\|_2}. \tag{9.10}$$

It is worth noting that, when using Algorithm 7, the maximum similarity in $\tilde{\mathbf{D}}$ is not hard-thresholded but rather indirectly controlled by the hyperparameter $\tilde{N}$.

9.3 Experiments

9.3.1 Simulation environment and baselines

Simulation settings

Two separate datasets are considered: a `DeepMIMO` [Alk19] dataset with the urban outdoor "O1" scenario where the downlink central frequency is set to 3.5GHz, and a `Sionna` [Hoy+22]

dataset generated with the *Etoile* scene in Paris at 28GHz. An illustration of these propagation environments is provided in Fig. 8.7. Additionally, both datasets considers $N_a = 64$ antennas and $N_s = 16$ subcarriers over a 20 MHz bandwidth, $N_c = 5$k calibration channels, 3k training channels and 2k test channels.

Note that, for all following simulations, the precoders are learned for only one subcarrier, e.g. the central subcarrier, but that it is possible to extend this approach to all subcarriers.

Hyperparameters and baselines

When considering encoder parameter subsampling, the number of kept channels is set to $\tilde{N} = 10$ unless stated otherwise. Similarly, the embedding space dimension is set to $d = 2$. For the decoder, $F = 200$ frequencies are considered, and σ_B is optimized for each dataset. Finally, the MLP layer size is set to $T = 128$.

Baselines consist of an MLP autoencoder where the embedding layer has dimension d as well as variants between the proposed approach and this MLP autoencoder: MLP encoder/RFF decoder and CC encoder/MLP decoder. Additionally, the RFF decoder with true locations as inputs is considered.

9.3.2 Results

Performance against baselines

It clearly appears in Fig. 9.4, that CC, as a model-based encoder, has a significant role in the mono-UE model performance: every model with a MLP encoder presents poor performance. Additionally, it can be seen that the RFF decoder architecture provides a slight increase in performance. As the location manifold is inherently bidimensional, it is not surprising that bidimensional chart locations perform well in the proposed approach. This holds true under the continuity hypothesis of the underlying location-to-channel mapping [Cha+24b; Cha+25a]. Additionally, note that the learned bidimensional channel charts carry more meaningful information than true locations, as the proposed architecture outperforms the RFF decoder used with true locations.

Encoder learning and subsampling impact

The different variants of the proposed approach are denoted as follows:

- (a): No encoder learning, no subsampling
- (b): Encoder learning, no subsampling
- (c): Encoder learning, subsampling ($\tilde{N} = 100$)
- (d): Encoder learning, subsampling ($\tilde{N} = 10$)

Table 9.1: Mono-UE precoder regression performance.

	Network	(a)	(b)	(c)	(d)
	Learnable parameters	101k	10.3M	306k	121k
`Sionna`	Median ρ (↑)	0.88	**0.94**	**0.94**	**0.94**
	Mean ρ (↑)	0.79	0.86	**0.88**	**0.88**
`DeepMIMO`	Median ρ (↑)	0.91	0.96	**0.97**	0.96
	Mean ρ (↑)	0.73	0.85	**0.87**	0.84

It is shown in Table 9.1 and Fig. 9.5 that allowing the encoder to be optimized during training is beneficial: both the median and mean normalized squared cosine similarities are increased. This implies that the channels and inferred precoders become more collinear, thereby enhancing communication performance. Additionally, it is shown that the subsampling procedure does not introduce a significant performance drop. Conversely, when considering $\tilde{N} = 100$, a slight performance increase is observed. The minor performance drop when considering $\tilde{N} = 10$ can be attributed to the possibility of reduced scene coverage with such limited number of calibration channels. Indeed, as new chart locations are inferred from a convex combination of the calibration chart locations, considering too few calibration channels may result in insufficient coverage for some regions of the chart location space. This can be seen in the region immediately on the left of the BS, denoted by a red cross, in Fig. 9.5. Notwithstanding, with the considered hyperparameters, the subsampling approach with $\tilde{N} = 10$ achieves almost a ninetyfold reduction in learnable parameters, with almost no loss in performance.

Compression ratio impact

The compression ratio can be defined as:

$$\gamma = \frac{2N_a N_s}{d+1}, \tag{9.11}$$

as downlink channel vectors are composed of $N_a N_s$ complex coefficients. The $+1$ term at the denominator arises from the requirement of channel norm transfer to the BS, as a denormalization step is performed.

Fig. 9.6 presents the evolution of the median squared cosine similarity of the proposed approach with respect to the compression ratio. This is achieved by varying the embedding space dimension d. It can be seen that increasing d, i.e. decreasing γ, slightly increases performance. It should also be noted that it is possible to reach good performance with $d = 2$, i.e. $\gamma \simeq 1024$, which significantly exceeds compression ratios reached with classical autoencoders, i.e. around 64. This confirms the interest of the proposed model-based approach.

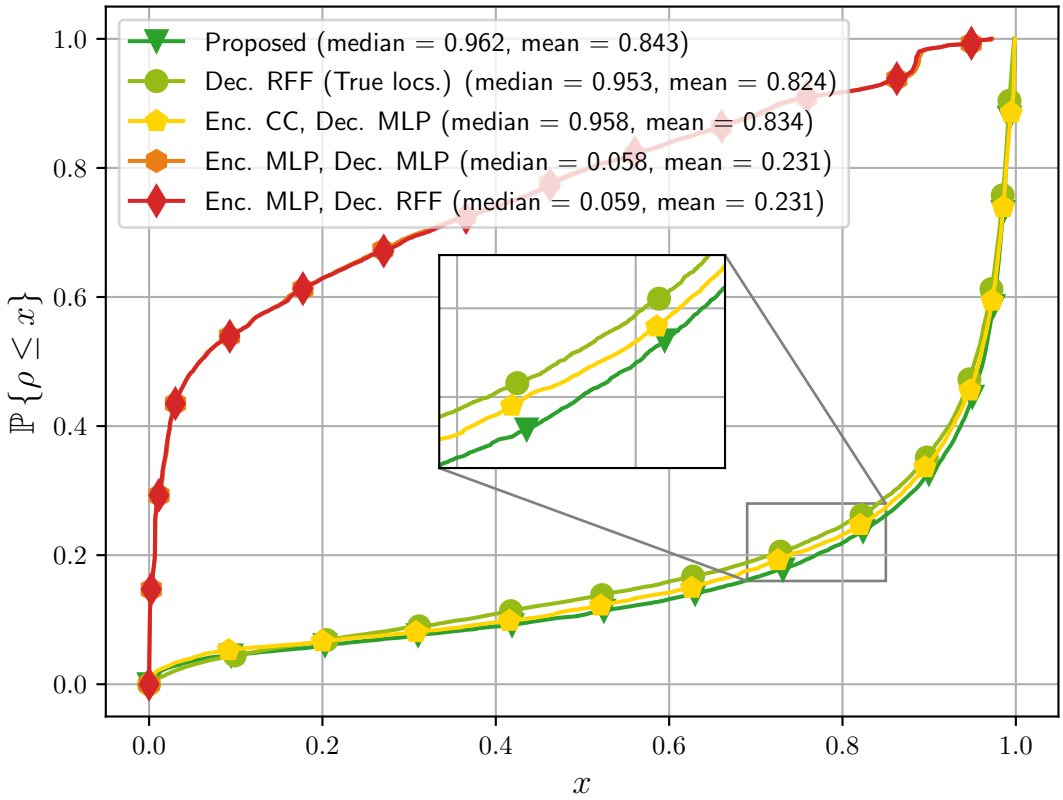


Figure 9.4: CDF of the squared cosine similarities, i.e. Eq. (9.4): `Sionna` dataset.

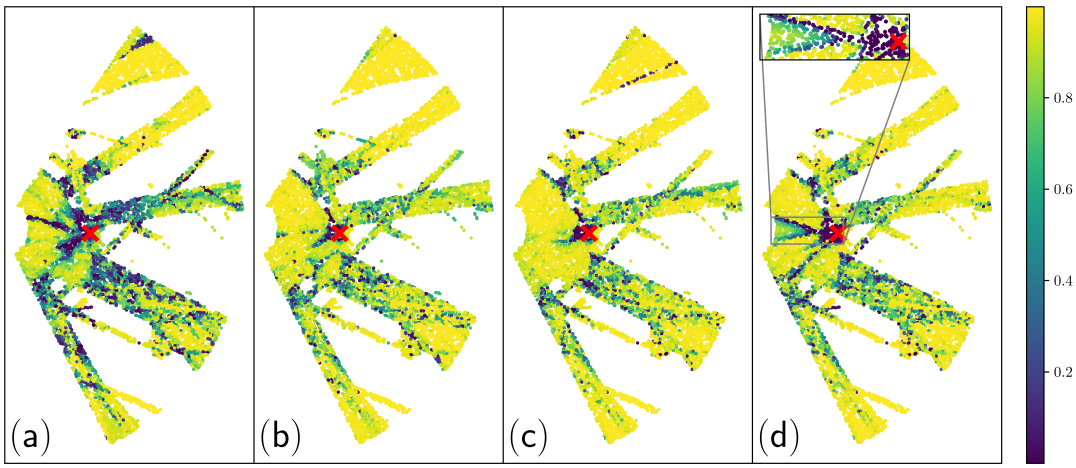


Figure 9.5: Squared cosine similarity map, i.e. Eq. (9.4): `Sionna` dataset.

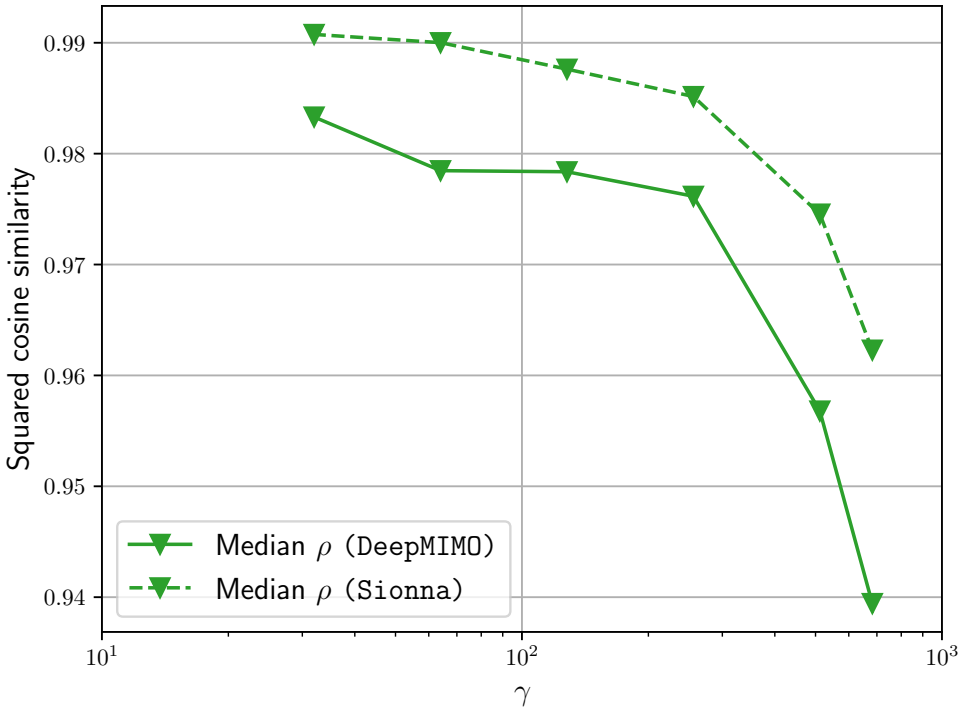


Figure 9.6: Median ρ evolution with the compression ratio.

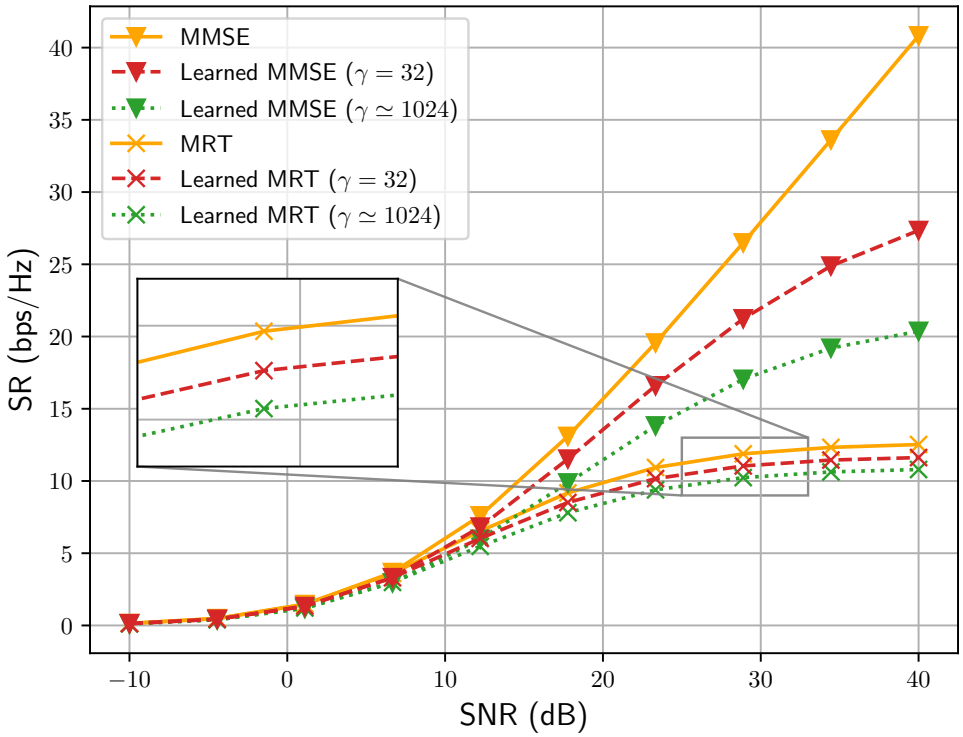


Figure 9.7: SR performance: `Sionna` dataset.

Muti-UE beamforming performance

Fig. 9.7 presents the multi-UE performance of the proposed approach. The MRT and MMSE SR curves are obtained by using true channel coefficients and computing Eq. (9.6) with the resulting MRT and MMSE precoders. The learned MRT and MMSE curves are obtained by concatenating the phase-shifted channel estimates obtained after the D_1 stage, and applying the MRT and MMSE precoder computations in the D_2 stage.

It can be observed in Fig. 9.7 that the learned MRT precoding matrix exhibits a behavior similar to that of the actual MRT precoder. Indeed, the learned MRT can be interpreted as the concatenation of mono-UE MRT precoders, each affected by a phase shift. Since this phase shift is canceled by the modulus operation in the SR computation [5], the slight performance degradation between the true and learned MRT precoders arises from the imperfect learning of the decoding function D_1. In contrast, this phase shift does not vanish when considering the MMSE precoder, thereby causing a more noticeable performance drop between the true and learned MMSE approaches. Nevertheless, the learned MMSE precoder still outperforms the actual MRT precoder and follows the same performance dynamics as the true MMSE precoder. This highlights the potential of the proposed approach, as it enables inter-UE interference cancellation through the proposed heuristic learning procedure, that only considers individual UEs and their compressed channel representations. Furthermore, as shown in Fig. 9.6, decreasing the compression ratio improves mono-UE precoder learning performance, which directly translates into enhanced inter-UE interference cancellation in the multi-UE setting.

9.4 Conclusion

This chapter tackled the problem of CSI compression in FDD systems following the MB-ML methodology, as formalized in Problem 3.4. Several remarks can be made, as presented below.

- The same encoder/decoder architectures as in Chapter 8 were used for the considered CSI compression problem. As such, domain knowledge, presented in (H1), has been used to **structure** and **initialize** the neural architectures of the encoder and decoder.

- The CSI compression problem was studied from a task-based perspective: instead of reconstructing CSI from a low-dimensional representation and subsequently using it to design a precoder, the proposed approach uses a neural architecture that directly infers a precoder from pseudo-locations obtained via channel charting.

- An end-to-end heuristic learning strategy was introduced to train the proposed architectures in an efficient manner. Moreover the number of learnable parameter of

5. See Eq. (9.6).

the proposed approach was optimized through a physically-grounded subsampling method.

- The proposed method was evaluated on realistic data, demonstrating superior performance over several baselines, particularly at high compression ratios.
- Overall, this study highlighted the effectiveness of channel charting as a CSI compression method for tasks beyond localization.

Conclusion and Perspectives

This chapter provides a brief summary of the results of this manuscript, and outlines some perspectives, categorized into short-term and long-term directions.

Contributions

Before delving into perspectives related to this thesis, a brief overview of the results is proposed.

Research axis (A1): Learning the location-to-channel mapping. This axis, addressed by Chapters 4 and 5, focused on the design of a neural architecture capable of learning the location-to-channel mapping, and on leveraging this model to enhance the performance of fingerprinting-based radio localization techniques.

In Chapter 4, it was shown that theoretical derivations from a physics-based propagation channel model could be exploited to structure a neural network tailored to this learning task. Experimental evaluations on realistic synthetic datasets, covering complex radio environments, showed that the proposed network achieved average reconstruction errors ranging from -20 dB to -40 dB. In contrast, conventional neural architectures, such as MLPs and RFFs, failed to learn the mapping, due to spectral bias limitations. Moreover, the proposed network was shown to implicitly encode the entire radio environment, achieving compression ratios up to 10^3 compared to storing all the channel coefficients over a location grid covering the considered radio environment, all while maintaining limited reconstruction error. Note that, due to the continuous nature of the learned mapping, this compression ratio is virtually infinite.

Chapter 5 demonstrated that this network could be used as a neural generative channel model, significantly improving the accuracy of traditional fingerprinting localization methods. In particular, the proposed approach achieved sub-wavelength localization accuracy, yielding localization errors between 10 and 10^{-3} cm across multiple complex radio environments. This corresponds to an improvement of up to two orders of magnitude compared to classical fingerprinting methods such as k-NN. Furthermore, the proposed method exhibited a tenfold reduction in memory requirements compared to k-NN, since, unlike conventional fingerprinting approaches, its performance does not depend on the size of a stored comparison dictionary.

Research axis (A2): Learning hardware impairments. This axis, covered by Chapters 6 and 7, addressed the joint estimation and compensation of hardware impairments in wireless systems. Chapter 6 focused on channel estimation under antenna gain and frequency offset impairments in the receiver chain, whereas Chapter 7 considered a DoA sensing scenario with impairments in the system's antenna positions and gains.

In Chapter 6, the inherent sparsity of the SISO-OFDM channel was exploited to

perform channel estimation through sparse recovery techniques. Following the approach of [YLM22], the matching pursuit algorithm was unfolded into the mpNet neural network architecture. Both its computational and sample complexity were optimized through hierarchical search and structured dictionary approaches. These improvements yielded a hundredfold reduction in the average inference time and a thousandfold reduction in the number of learnable parameters, while achieving superior channel estimation accuracy compared to both the original mpNet and classical methods.

Chapter 7 built on the observation that the classical MUSIC algorithm, widely used in DoA estimation problems, is non-differentiable, making it unsuitable for jointly estimating DoAs and hardware impairments using first order techniques. To overcome this limitation, a differentiable variant of MUSIC was proposed, along with task-specific supervised and unsupervised loss functions. Simulation results showed that this differentiable framework enabled effective compensation of significant antenna gain and position errors, while simultaneously improving DoA estimation accuracy. Specifically, the proposed approach achieved a 99% reduction in the average angular estimation error with one source, and a 58% reduction with five sources, compared to the traditional MUSIC algorithm without hardware impairments knowledge.

Research axis (A3): Compressing the channel. This axis, presented in Chapters 8 and 9, investigated the use of channel charting for the efficient compression of channel coefficients. As modern wireless communication systems operate with increasingly larger antenna arrays and over numerous subcarriers, the dimensionality of the channel has grown substantially. Such high dimensionality results in considerable overhead for communication tasks that depend on accurate channel knowledge. To overcome this limitation, this work explored the use of channel charting in two specific contexts: beam prediction in cell-free massive MIMO systems, and precoder prediction in FDD massive MIMO systems.

In Chapter 8, the objective was to design a procedure for selecting the optimal beam in a predefined codebook, to serve a given UE across multiple BSs operating over distinct uplink and downlink frequencies, in a cell-free setting. After a single BS transmitted pilot signals to the UE, channel charting was applied to reduce the dimensionality of the channel information fed back by the UE. The resulting low-dimensional representation was then shared among BSs in the cell-free network, and each BS used a dedicated neural network to map this representation to the index of the optimal beam within the codebook. The proposed approach yielded notable performance gains, achieving a 3 to 12% improvement in the top-3 beam selection accuracy compared to a traditional MLP decoder. Furthermore, the low-dimensional channel representations obtained through channel charting were shown to be sufficiently informative for simple decoding methods such as k-NN to achieve satisfactory performance, i.e. between 78 and 91% for the top-3 beam selection accuracy.

Chapter 9 focused on task-oriented CSI compression in FDD massive MIMO systems. In this setting, the feedback information transmitted from the UE to the BS was compressed into a low-dimensional representation through channel charting, at the UE level, which was then directly decoded into a precoding matrix using a neural network,

at the BS level. A heuristic learning strategy and a physically-grounded subsampling of the encoder's learnable parameters were used to optimize both the time and parameter complexity of the proposed encoder and decoder. Simulation results showed that the proposed method achieved excellent beamforming performance at very high compression ratios, approximately 1024, compared to 64 in conventional CSI compression schemes. Moreover, the subsampling strategy enabled an eightyfold reduction in the number of learnable parameters, with negligible degradation in beamforming performance relative to the non-subsampled baseline.

Perspectives

The contributions of this thesis could be further refined by exploring multiple improvement directions, categorized into short-term and long-term research perspectives.

Short term

Each contribution entails specific challenges and limitations, which can be grouped into three problematics, as discussed below.

(P1) **Data quality.** All the contributions presented in this thesis were evaluated using realistic synthetic datasets. While such data serve as a valuable surrogate for actual measurements, they inevitably lack the imperfections and uncertainties inherent to real-world communication and sensing systems. Consequently, evaluating the proposed approaches on real measurements would help identify the adaptations required to ensure their robustness and effectiveness in real-world conditions.

(P2) **Reliance on simplifying assumptions.** The contributions presented in this thesis rely on simplifying assumptions within the considered system models, which may limit their applicability in real-world scenarios. Specifically, Chapters 4 and 5 assume a static or quasi-static propagation environment, Chapter 6 focuses exclusively on mono-antenna systems, Chapter 7 only considers uncorrelated sources, and Chapters 8 and 9 assume perfect time synchronization between UEs and BSs.

(P3) **Method refinements.** Although the methods proposed in this thesis demonstrate satisfactory performance, their efficiency could be further improved by reducing the number of learnable parameters or optimizing their computational complexity.

Several potential directions for improvement can be identified to address these challenges, as outlined below.

- **Physical testbed.** For all contributions, problematic (P1) is both the most straightforward to define and the most difficult to address. Indeed, obtaining realistic data requires the deployment of a physical testbed, which entails numerous engineering

constraints, including synchronization, hardware design, compensation for hardware impairments, and high-level system orchestration. Moreover, constructing a testbed capable of replicating a cell-free massive MIMO system constitutes a particularly challenging task.

- **3D locations.** In Chapter 4, in order to reduce the computational complexity of the simulations, the performance of the proposed neural network was assessed only for locations belonging to the same elevation plane. Extending the simulations to locations in $\mathbb{R}^3$, thereby addressing Problematic (P3), requires sampling spatial frequencies from the unit sphere. Since achieving the same coverage on a sphere as on a circle necessitates a much larger number of samples, the number of learnable parameters of the proposed method could be reduced by optimizing a smaller set of spatial frequencies, for instance by learning them. However, to prevent the initialized spatial frequencies from converging towards the dominant ones, potentially causing an incomplete representation of highly attenuated NLoS paths, a spherical validity domain could be defined for each learnable spatial frequency, thereby ensuring a balanced representation of all regions of the unit sphere.

- **Online learning.** In Chapters 4 and 5, the primary simplifying assumption lies in the consideration of static or quasi-static propagation environments, which does not accurately reflect all radio scenarios. For the contributions presented in these chapters, problematic (P2) could be addressed by adopting an online learning strategy. Specifically, the proposed location-to-channel network would initially be trained on a synthetic dataset, as presented in this thesis, and then continuously updated with periodically collected channel measurements, allowing the network to adapt to dynamic environmental variations.

- **Multidimensional mpNet.** In Chapter 6, extending the system model to multi-antenna, multi-frequency scenarios represents a promising direction. In principle, mpNet could be applied in such settings by constructing a composite dictionary of frequency response vectors and steering vectors. However, this approach would incur intractable complexity due to the excessive number of required atoms. A more practical alternative is to adopt a bi-level strategy, in which, for a given noisy channel, the most correlated steering vector is first selected at each iteration, followed by a subsequent selection among the frequency response vectors. This can be interpreted as the unfolded implementation of the multidimensional OMP algorithm presented in [PGPR22]. This strategy would substantially reduce the computational complexity while preserving the benefits of the sparse recovery framework, thereby jointly addressing (P2) and (P3).

- **Spatial augmentation.** The primary limitation of Chapter 7 stems from the assumption of uncorrelated sources. As such, addressing (P2), could be achieved by employing the spatial augmentation technique presented in [Ge+24] and originally

introduced in [Gon+22]. This method restores the rank condition of the measurement matrix, which is typically not satisfied in the presence of correlated sources. Specifically, it leverages measurements across multiple frequencies and introduces computational steps that transfer measurement resources from the frequency to the antenna domain.

- **True CSI compression.** Chapter 9 presented a task-oriented method for decoding compressed channel coefficients into a precoding matrix. While this approach demonstrated strong performance, directly decoding the true channel coefficients from their compressed representation would account for a broader class of applications. As shown in Eq. (9.8), minimizing the proposed loss function results in decoding the compressed channels into normalized channels affected by a specific phase shift. Consequently, addressing (P3) could be achieved by transmitting a phase reference from the UE alongside the compressed channel representation, and using it at the BS to counter the aforementioned phase shift, thereby recovering the true channel coefficients after decoding.

- **Channel charting based UE grouping policy.** In Chapter 9, the performance of the proposed method is evaluated by computing the average sum-rate over B groups of K randomly selected UEs. However, this approach may encounter pathological cases in which all UEs are closely located, making the task of inferring a precoding matrix from pseudo-locations particularly challenging. As a result, a UE grouping strategy based on the pseudo-locations could be used to address (P3). Specifically, since channel charting aims to preserve spatial neighborhoods, pseudo-locations that are close in the representation space are expected to correspond to physically close UEs. Therefore, by constructing a distance graph between pseudo-locations, the BS could form UE groups with maximized inter-UE distances, thereby improving beamforming performance.

Long term

In addition to the aforementioned short-term perspectives, this thesis has brought to light several new open questions that would require dedicated studies. These are presented below.

- **Linking MB-ML and NTK.** As discussed in Highlight 4.1, the NTK theory provides a powerful framework for analyzing the theoretical learning performance of neural architectures. Specifically, it establishes that neural networks can more effectively learn functions that lie within the span of their strongest NTK eigenfunctions. Although this theory is formally valid in the infinite-width regime, it has been shown to offer a reasonably accurate surrogate for learnability in the practical finite-width setting. Within the MB-ML paradigm, one approach consists in leveraging domain knowledge to structure neural architectures, thereby introducing problem-specific

inductive biases. In this context, it would be valuable to investigate how incorporating such domain knowledge influences the NTK of the resulting structured architectures, and whether theoretical learning guarantees can be established under these conditions.

- **Optimizing training dataset size.** As presented in Chapters 4 and 5, a limitation of the proposed network for learning the location-to-channel mapping lies in the size of the training dataset. In this thesis, the dataset was generated by randomly sampling the location space associated with the considered radio environment. While it was shown that reducing the spatial sampling density did not significantly degrade performance up to a certain point, it would be interesting to investigate whether a specific sampling policy could further optimize the dataset size. More precisely, one could consider combining sufficiently spaced locations, to capture the diversity of propagation paths across the radio environment, with closely spaced locations, to account for small-scale channel characteristics.

- **Imaging the propagation environment.** In Chapters 4 and 5, it was shown that the proposed neural architecture for learning the location-to-channel mapping effectively captured the underlying propagation physics in the considered environments. This was achieved using only a dataset of locations and their associated channel coefficients, without any additional a priori information about the propagation scene. The trained model could be leveraged for imaging purposes within the same propagation environment. Specifically, for a given location within the environment, the trained model provides information about the activated planar wavefronts along with their corresponding activation coefficients. This information could subsequently be integrated with source separation methods to localize both physical and virtual sources within the scene, thereby enabling the identification of reflecting and diffracting objects.

- **Linking channel charting and source coding.** As discussed in Chapters 8 and 9, CSI compression consists in encoding channel coefficients into a low-dimensional latent space and subsequently decoding them to reconstruct the original channel. Recent advances in this field involve the joint optimization of encoder/decoder neural networks along with a source coding oriented model. The latter leverages an entropy model to estimate the distribution of the encoded channels, enabling the evaluation of the entropy bit size of the latent representation, i.e. the achievable bit size using an entropy coding scheme, such as arithmetic coding, that uses the estimated distribution by the entropy model. In this framework, the learning objective typically combines a reconstruction loss (associated with the encoder/decoder pair), and an entropy-related term accounting for the entropy bit size (source coding efficiency). An interested reader is referred to [PDL25] for more information. It would be particularly interesting to consider this approach with a channel charting based encoder, to confirm that the very low dimensionality of its latent space, i.e. $\mathbb{R}^2$ in Chapter 9, indeed yields a low entropy bit size.

Part III

Appendices and bibliography

Appendix A

Chapter 1's appendix

A.1 Proof of Eq. (1.2)

Proof. The required clearance is obtained as the maximum of the semi-minor axis length of the Fresnel ellipsoid. As this ellipsoid is defined from the condition:

$$d_1(\mathbf{x}) + d_2(\mathbf{x}) - D \leq \frac{\lambda}{2}, \tag{A.1}$$

the required clearance is obtained by setting $d_1(\mathbf{x}) = d_2(\mathbf{x}) = d$, and considering that $\mathbf{x}$ lies on the boundary of the ellipsoid, which yields:

$$2d - D = \frac{\lambda}{2}. \tag{A.2}$$

From Fig. 1.2, one can remark that $d^2 = (D/2)^2 + r_{\mathsf{cl}}^2$, which can be developed as:

$$2\sqrt{\left(\frac{D}{2}\right)^2 + r_{\mathsf{cl}}^2} - D = \frac{\lambda}{2} \tag{A.3}$$

$$\Leftrightarrow \frac{D^2}{4} + r_{\mathsf{cl}}^2 = \frac{D^2}{4} + \frac{\lambda D}{4} + \frac{\lambda^2}{16} \tag{A.4}$$

$$\Leftrightarrow r_{\mathsf{cl}} = \sqrt{\frac{\lambda^2}{16} + \frac{\lambda D}{4}} \tag{A.5}$$

$$\overset{(*)}{\Leftrightarrow} r_{\mathsf{cl}} \simeq \frac{1}{2}\sqrt{\lambda D}, \tag{A.6}$$

where $(*)$ is obtained by recalling that $\lambda = c/f \ll D$ in usual communication systems, yielding:

$$\lambda \ll D \tag{A.7}$$

$$\Rightarrow \lambda^2 \ll \lambda D \tag{A.8}$$

$$\Rightarrow \frac{\lambda^2}{16} \ll \frac{\lambda D}{16} < \frac{\lambda D}{4}. \tag{A.9}$$

■

A.2 Proof of Definition 1.2

Proof. By definition [1]:

$$h(t,f) = \mathcal{F}_\tau\left[h(t,\tau)\right] \tag{A.10}$$

$$= \int_{\mathbb{R}^+} h(t,\tau)\,\mathrm{e}^{-\mathrm{j}2\pi\tau f}\mathrm{d}\tau \tag{A.11}$$

$$= \int_{\mathbb{R}^+} \sum_{l=1}^{L_p} \alpha_l(t)\,\delta(\tau-\tau_l(t))\,\mathrm{e}^{-\mathrm{j}2\pi f_{d,l}t}\mathrm{e}^{-\mathrm{j}2\pi\tau f}\mathrm{d}\tau \tag{A.12}$$

$$= \sum_{l=1}^{L_p} \alpha_l(t) \int_{\mathbb{R}^+} \delta(\tau-\tau_l(t))\,\mathrm{e}^{-\mathrm{j}2\pi f_{d,l}t}\mathrm{e}^{-\mathrm{j}2\pi\tau f}\mathrm{d}\tau \tag{A.13}$$

$$= \sum_{l=1}^{L_p} \alpha_l(t)\,\mathrm{e}^{-\mathrm{j}2\pi f\tau_l(t)}\mathrm{e}^{-\mathrm{j}2\pi f_{d,l}t}, \tag{A.14}$$

which concludes the proof. ■

A.3 Proof of Eq. (1.30)

Proof. Assuming a MIMO system with N_r receive antennas and N_t transmit antennas, considering a uniform power allocation between all the transmitter antennas translates as:

$$\mathbf{Q} = \frac{P_e}{N_t}\mathbf{Id}_{N_t}, \tag{A.15}$$

P_e being the total emitted power. The SVD decomposition of the channel matrix $\mathbf{H} \in \mathbb{C}^{N_r\times N_t}$ reads as:

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{H}}, \tag{A.16}$$

with $\mathbf{U} \in \mathbb{C}^{N_r\times N_r}$ and $\mathbf{V} \in \mathbb{C}^{N_t\times N_t}$ being unitary, and $\mathbf{\Sigma} \in \mathbb{R}^{N_r\times N_t}$ being rectangular diagonal. Based on Definition 1.9, the MIMO capacity can rewritten as:

$$\mathsf{C}_{\mathrm{MIMO}} = B\log_2\left(\det\left(\mathbf{Id}_{N_r} + \frac{1}{\sigma_n^2}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{H}}\right)\right) \tag{A.17}$$

$$= B\log_2\left(\det\left(\mathbf{Id}_{N_r} + \frac{P_e}{N_t\sigma_n^2}\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{H}}\right)\mathbf{Id}_{N_t}\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{H}}\right)^{\mathsf{H}}\right)\right) \tag{A.18}$$

$$= B\log_2\left(\det\left(\mathbf{Id}_{N_r} + \frac{P_e}{N_t\sigma_n^2}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{H}}\mathbf{V}\mathbf{\Sigma}^{\mathsf{H}}\mathbf{U}^{\mathsf{H}}\right)\right) \tag{A.19}$$

$$= B\log_2\left(\det\left(\mathbf{Id}_{N_r} + \frac{P_e}{N_t\sigma_n^2}\mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^{\mathsf{H}}\mathbf{U}^{\mathsf{H}}\right)\right). \tag{A.20}$$

1. Note that due to the strictly positive nature of delays, the Fourier transform is only taken along the $\mathbb{R}^+$ space.

Leveraging the identity $\det(\mathbf{Id} + \mathbf{AB}) = \det(\mathbf{Id} + \mathbf{BA})$ yields:

$$\mathsf{C}_{\text{MIMO}} = B \log_2 \left(\det \left(\mathbf{Id}_{N_r} + \frac{P_e}{N_t \sigma_n^2} \mathbf{\Sigma}^{\mathsf{H}} \mathbf{U}^{\mathsf{H}} \mathbf{U} \mathbf{\Sigma} \right) \right) \tag{A.21}$$

$$= B \log_2 \left(\det \left(\mathbf{Id}_{N_r} + \frac{P_e}{N_t \sigma_n^2} \mathbf{\Sigma}^{\mathsf{H}} \mathbf{\Sigma} \right) \right). \tag{A.22}$$

Based on the SVD properties, one knows that the diagonal of $\mathbf{\Sigma}$ holds the square root of the eigenvalues of $\mathbf{HH}^{\mathsf{H}}$, e.g. considering $N_r > N_t$:

$$\mathbf{\Sigma} = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\lambda_r} \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.23}$$

where $r = \min(N_r, N_t) = \text{rk}(\mathbf{H})$. $\mathbf{\Sigma}$ being rectangular diagonal, $\mathbf{\Sigma}^{\mathsf{H}}\mathbf{\Sigma} \in \mathbb{R}^{N_r \times N_r}$ is diagonal. Then, the determinant of a diagonal matrix being the product of its diagonal entries, one obtains:

$$\mathsf{C}_{\text{MIMO}} = B \log_2 \left(\prod_{i=1}^{\text{rk}(\mathbf{H})} \left(1 + \frac{P_e}{N_t \sigma_n^2} \lambda_i \right) \right) \tag{A.24}$$

$$= B \sum_{i=1}^{\text{rk}(\mathbf{H})} \log_2 \left(1 + \frac{P_e}{N_t \sigma_n^2} \lambda_i \right), \tag{A.25}$$

which concludes the proof. ■

A.4 Proof of η formulation in Eq. (1.53)

The following inversion lemma is needed for the following derivations.

Lemma A.1: Hermitian of a matrix inverse

Let $\mathbf{A}$ be an invertible matrix. Then:

$$\left(\mathbf{A}^{-1}\right)^{\mathsf{H}} = \left(\mathbf{A}^{\mathsf{H}}\right)^{-1}. \tag{A.26}$$

The proof of this lemma follows directly by considering:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{Id} \Leftrightarrow \left(\mathbf{A}^{-1}\mathbf{A}\right)^{\mathsf{H}} = \mathbf{Id}^{\mathsf{H}} \tag{A.27}$$

$$\Leftrightarrow \mathbf{A}^{\mathsf{H}}\left(\mathbf{A}^{-1}\right)^{\mathsf{H}} = \mathbf{Id} \tag{A.28}$$

$$\Leftrightarrow \left(\mathbf{A}^{-1}\right)^{\mathsf{H}} = \left(\mathbf{A}^{\mathsf{H}}\right)^{-1}. \tag{A.29}$$

Proof. The goal is to derive the normalization factor η for the ZF precoder. Recalling than this precoder expresses as $\mathbf{W} = \mathbf{H}\left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1}$, one obtains:

$$\mathbf{W}^{\mathsf{H}}\mathbf{W} = \left(\mathbf{H}\left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1}\right)^{\mathsf{H}}\left(\mathbf{H}\left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1}\right) \tag{A.30}$$

$$= \left(\left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1}\right)^{\mathsf{H}}\mathbf{H}^{\mathsf{H}}\mathbf{H}\left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1} \tag{A.31}$$

$$= \left(\left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1}\right)^{\mathsf{H}} \tag{A.32}$$

$$\overset{(*)}{=} \left(\left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{\mathsf{H}}\right)^{-1} \tag{A.33}$$

$$= \left(\mathbf{H}^{\mathsf{H}}\mathbf{H}\right)^{-1}, \tag{A.34}$$

where $(*)$ originates from Lemma A.1. Finally, the normalization factor of the ZF precoder can be expressed as:

$$\eta = \frac{1}{\sqrt{\mathrm{Tr}\left(\mathbf{W}^{\mathsf{H}}\mathbf{W}\right)}} \tag{A.35}$$

$$= \frac{1}{\sqrt{\mathrm{Tr}\left((\mathbf{H}^{\mathsf{H}}\mathbf{H})^{-1}\right)}}, \tag{A.36}$$

which concludes the proof. ■

Appendix B

Chapter 2's appendix

B.1 Proof of Eq. (2.27)

Proof. The objective is to solve:

$$\begin{aligned} \underset{\mathbf{W}}{\text{minimize}} \quad & \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \|\mathbf{y} - \mathbf{W}\mathbf{x}\|_2^2 , \\ \text{subject to} \quad & \mathbf{W} \in \mathbb{R}^{m\times l} \end{aligned} \tag{B.1}$$

As the considered loss function, i.e. the ℓ_2 distance, is convex, an exact solution can be obtained by finding the optimal weight matrix $\mathbf{W}^\star$, for which the gradient of the loss function vanishes. Introducing $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N] \in \mathbb{R}^{m\times N}$, and $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{l\times N}$, the minimization objective can be rewritten as:

$$\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \|\mathbf{y} - \mathbf{W}\mathbf{x}\|_2^2 = \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_\mathsf{F}^2 \tag{B.2}$$

$$= \mathrm{Tr}\left((\mathbf{Y} - \mathbf{W}\mathbf{X})^\mathsf{T} (\mathbf{Y} - \mathbf{W}\mathbf{X})\right) \tag{B.3}$$

$$= \mathrm{Tr}\left(\mathbf{Y}^\mathsf{T}\mathbf{Y}\right) - \mathrm{Tr}\left(\mathbf{Y}^\mathsf{T}\mathbf{W}\mathbf{X}\right) - \mathrm{Tr}\left(\mathbf{X}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{Y}\right) \tag{B.4}$$

$$+ \mathrm{Tr}\left(\mathbf{X}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{W}\mathbf{X}\right).$$

Recalling that:

$$\mathrm{Tr}\left(\mathbf{A}^\mathsf{T}\mathbf{B}\right) = \mathrm{Tr}\left(\mathbf{B}\mathbf{A}^\mathsf{T}\right) \tag{B.5}$$

$$\nabla_\mathbf{W}\mathrm{Tr}\left(\mathbf{A}\mathbf{W}^\mathsf{T}\mathbf{B}\right) = \mathbf{B}\mathbf{A} \tag{B.6}$$

$$\nabla_\mathbf{W}\mathrm{Tr}\left(\mathbf{A}\mathbf{W}\mathbf{B}\right) = \mathbf{A}^\mathsf{T}\mathbf{B}^\mathsf{T} \tag{B.7}$$

$$\nabla_\mathbf{W}\mathrm{Tr}\left(\mathbf{W}^\mathsf{T}\mathbf{W}\mathbf{A}\right) = \mathbf{W}\mathbf{A}^\mathsf{T} + \mathbf{W}\mathbf{A}, \tag{B.8}$$

the gradient of the loss function reads as:

$$\nabla_\mathbf{W}\frac{1}{|\mathcal{D}|}\|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_\mathsf{F}^2 = \frac{2}{|\mathcal{D}|}\left(\mathbf{W}\mathbf{X}\mathbf{X}^\mathsf{T} - \mathbf{Y}\mathbf{X}^\mathsf{T}\right). \tag{B.9}$$

Indeed, one has:

$$\nabla_{\mathbf{W}} \mathrm{Tr}\left(\mathbf{Y}^{\mathsf{T}}\mathbf{W}\mathbf{X}\right) = \mathbf{Y}\mathbf{X}^{\mathsf{T}} \tag{B.10}$$

$$\nabla_{\mathbf{W}} \mathrm{Tr}\left(\mathbf{X}^{\mathsf{T}}\mathbf{W}^{\mathsf{T}}\mathbf{Y}\right) = \mathbf{Y}\mathbf{X}^{\mathsf{T}}, \tag{B.11}$$

and

$$\nabla_{\mathbf{W}} \mathrm{Tr}\left(\mathbf{X}^{\mathsf{T}}\mathbf{W}^{\mathsf{T}}\mathbf{W}\mathbf{X}\right) = \nabla_{\mathbf{W}} \mathrm{Tr}\left(\mathbf{W}^{\mathsf{T}}\mathbf{W}\mathbf{X}\mathbf{X}^{\mathsf{T}}\right) \tag{B.12}$$

$$= \mathbf{W}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{\mathsf{T}} + \mathbf{W}\mathbf{X}\mathbf{X}^{\mathsf{T}} \tag{B.13}$$

$$= 2\mathbf{W}\mathbf{X}\mathbf{X}^{\mathsf{T}}. \tag{B.14}$$

The following expression is finally obtained:

$$\nabla_{\mathbf{W}} \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \|\mathbf{y} - \mathbf{W}\mathbf{x}\|_2^2 = 0 \tag{B.15}$$

$$\Leftrightarrow \frac{2}{|\mathcal{D}|}\left(\mathbf{W}^{\star}\mathbf{X}\mathbf{X}^{\mathsf{T}} - \mathbf{Y}\mathbf{X}^{\mathsf{T}}\right) = 0 \tag{B.16}$$

$$\Leftrightarrow \mathbf{W}^{\star}\mathbf{X}\mathbf{X}^{\mathsf{T}} = \mathbf{Y}\mathbf{X}^{\mathsf{T}} \tag{B.17}$$

$$\Leftrightarrow \mathbf{W}^{\star} = \mathbf{Y}\mathbf{X}^{\mathsf{T}}\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)^{-1}. \tag{B.18}$$

■

B.2 Proof of Eq. (2.33)

Proof. The objective is to solve:

$$\begin{aligned} \underset{\mathbf{V}}{\text{maximize}} \quad & \frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \left\|\mathbf{V}^{\mathsf{T}}\mathbf{x}\right\|_2^2. \\ \text{subject to} \quad & \mathbf{V} \in \mathbb{R}^{D\times d}, \\ & \mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbf{Id}_d \end{aligned} \tag{B.19}$$

Introducing $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$, the maximization objective rewrites as:

$$\sum_{\mathbf{x}\in\mathcal{D}} \left\|\mathbf{V}^{\mathsf{T}}\mathbf{x}\right\|_2^2 = \left\|\mathbf{V}^{\mathsf{T}}\mathbf{X}\right\|_{\mathsf{F}}^2 \tag{B.20}$$

$$= \mathrm{Tr}\left(\mathbf{X}^{\mathsf{T}}\mathbf{V}\mathbf{V}^{\mathsf{T}}\mathbf{X}\right) \tag{B.21}$$

$$\overset{(*)}{=} \mathrm{Tr}\left(\mathbf{V}^{\mathsf{T}}\mathbf{X}\mathbf{X}^{\mathsf{T}}\mathbf{V}\right), \tag{B.22}$$

where $(*)$ originates from the circularity property of the trace operator. The optimization

problem then rewrites as:

$$\begin{aligned} \underset{\mathbf{V}}{\text{maximize}} \quad & \frac{1}{|\mathcal{D}|}\mathrm{Tr}\left(\mathbf{V}^\mathsf{T}\mathbf{X}\mathbf{X}^\mathsf{T}\mathbf{V}\right), \\ \text{subject to} \quad & \mathbf{V} \in \mathbb{R}^{D\times d}, \\ & \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{Id}_d \end{aligned} \tag{B.23}$$

which can be solved using a Lagrangian method. Indeed, introducing a diagonal matrix of Lagrange multipliers $\mathbf{\Lambda}_\mathcal{L} \in \mathbb{R}^{d\times d}$, the Lagrangian writes as:

$$\mu\left(\mathbf{V}, \mathbf{\Lambda}_\mathcal{L}\right) = \mathrm{Tr}\left(\mathbf{V}^\mathsf{T}\mathbf{X}\mathbf{X}^\mathsf{T}\mathbf{V}\right) - \mathrm{Tr}\left(\mathbf{\Lambda}_\mathcal{L}\left(\mathbf{V}^\mathsf{T}\mathbf{V} - \mathbf{Id}_d\right)\right), \tag{B.24}$$

and a solution to the optimization described in Eq. (B.23) satisfies:

$$\nabla_{\mathbf{\Lambda}_\mathcal{L}}\mu\left(\mathbf{V}, \mathbf{\Lambda}_\mathcal{L}\right) = \mathbf{0} \tag{B.25}$$

$$\nabla_{\mathbf{V}}\mu\left(\mathbf{V}, \mathbf{\Lambda}_\mathcal{L}\right) = \mathbf{0}. \tag{B.26}$$

Gradient computation wrt. $\mathbf{\Lambda}_\mathcal{L}$. Recalling that:

$$\nabla_{\mathbf{V}}\mathrm{Tr}\left(\mathbf{V}\mathbf{A}\right) = \mathbf{A}^\mathsf{T}, \tag{B.27}$$

the Lagrangian gradients wrt. $\mathbf{\Lambda}_\mathcal{L}$ can be evaluated as follows:

$$\begin{aligned} \nabla_{\mathbf{\Lambda}_\mathcal{L}}\mu\left(\mathbf{V}, \mathbf{\Lambda}_\mathcal{L}\right) &= \nabla_{\mathbf{\Lambda}_\mathcal{L}}\mathrm{Tr}\left(\mathbf{V}^\mathsf{T}\mathbf{X}\mathbf{X}^\mathsf{T}\mathbf{V}\right) - \nabla_{\mathbf{\Lambda}_\mathcal{L}}\mathrm{Tr}\left(\mathbf{\Lambda}_\mathcal{L}\left(\mathbf{V}^\mathsf{T}\mathbf{V} - \mathbf{Id}_d\right)\right) && \text{(B.28)} \\ &= -\nabla_{\mathbf{\Lambda}_\mathcal{L}}\mathrm{Tr}\left(\mathbf{\Lambda}_\mathcal{L}\left(\mathbf{V}^\mathsf{T}\mathbf{V} - \mathbf{Id}_d\right)\right) && \text{(B.29)} \\ &= -\left(\mathbf{V}^\mathsf{T}\mathbf{V} - \mathbf{Id}_d\right)^\mathsf{T} && \text{(B.30)} \\ &= -\left(\mathbf{V}^\mathsf{T}\mathbf{V} - \mathbf{Id}_d\right). && \text{(B.31)} \end{aligned}$$

The gradient vanishing condition thus yields:

$$\mathbf{V}^{\star\mathsf{T}}\mathbf{V}^\star = \mathbf{Id}_d, \tag{B.32}$$

which ensures that the stationary Lagrangian point respects the problem's constraint.

Gradient computation wrt. $\mathbf{V}$. Recalling that:

$$\nabla_{\mathbf{V}}\mathrm{Tr}\left(\mathbf{V}\mathbf{V}^\mathsf{T}\mathbf{A}\right) = \mathbf{A}\mathbf{V} + \mathbf{A}^\mathsf{T}\mathbf{V} \tag{B.33}$$

$$\nabla_{\mathbf{V}}\mathrm{Tr}\left(\mathbf{V}^\mathsf{T}\mathbf{V}\mathbf{A}\right) = \mathbf{V}\mathbf{A}^\mathsf{T} + \mathbf{V}\mathbf{A}, \tag{B.34}$$

the Lagrangian gradients wrt. $\mathbf{V}$ can be evaluted as follows:

$$\nabla_{\mathbf{V}}\mu\left(\mathbf{V}, \mathbf{\Lambda}_\mathcal{L}\right) = \nabla_{\mathbf{V}}\mathrm{Tr}\left(\mathbf{V}^\mathsf{T}\mathbf{X}\mathbf{X}^\mathsf{T}\mathbf{V}\right) - \nabla_{\mathbf{V}}\mathrm{Tr}\left(\mathbf{\Lambda}_\mathcal{L}\left(\mathbf{V}^\mathsf{T}\mathbf{V} - \mathbf{Id}_d\right)\right) \tag{B.35}$$

$$\overset{(*)}{=} \nabla_{\mathbf{V}} \mathrm{Tr}\left(\mathbf{V}\mathbf{V}^{\mathsf{T}}\mathbf{X}\mathbf{X}^{\mathsf{T}}\right) - \nabla_{\mathbf{V}} \mathrm{Tr}\left(\mathbf{V}^{\mathsf{T}}\mathbf{V}\mathbf{\Lambda}_{\mathcal{L}}\right) \tag{B.36}$$

$$= \mathbf{X}\mathbf{X}^{\mathsf{T}}\mathbf{V} + \mathbf{X}\mathbf{X}^{\mathsf{T}}\mathbf{V} - \mathbf{V}\mathbf{\Lambda}_{\mathcal{L}}^{\mathsf{T}} - \mathbf{V}\mathbf{\Lambda}_{\mathcal{L}} \tag{B.37}$$

$$\overset{(**)}{=} 2\mathbf{X}\mathbf{X}^{\mathsf{T}}\mathbf{V} - 2\mathbf{V}\mathbf{\Lambda}_{\mathcal{L}}, \tag{B.38}$$

where $(*)$ originates from the circularity of the trace operator, and $(**)$ from diagonality of $\mathbf{\Lambda}_{\mathcal{L}}$. One then obtains:

$$\nabla_{\mathbf{V}} \mu\left(\mathbf{V}, \mathbf{\Lambda}_{\mathcal{L}}\right) = \mathbf{0} \tag{B.39}$$

$$\Leftrightarrow \mathbf{X}\mathbf{X}^{\mathsf{T}}\mathbf{V}^{\star} = \mathbf{V}^{\star}\mathbf{\Lambda}_{\mathcal{L}}, \tag{B.40}$$

which, recalling that $\mathbf{V}^{\star} \in \mathbb{R}^{D\times d}$, with $d < D$, describes a matrix eigenvalue problem, where $\mathbf{V}^{\star}$ contains d eigenvectors of $\mathbf{X}\mathbf{X}^{\mathsf{T}}$ and $\mathbf{\Lambda}_{\mathcal{L}}$ contains the corresponding eigenvalues on its diagonal. Maximizing the objective of the problem described in Eq. (B.23), is then obtained by taking $\mathbf{V}^{\star}$ in the trace:

$$\mathrm{Tr}\left(\mathbf{V}^{\star^{\mathsf{T}}}\mathbf{X}\mathbf{X}^{\mathsf{T}}\mathbf{V}^{\star}\right) = \mathrm{Tr}\left(\mathbf{V}^{\star^{\mathsf{T}}}\mathbf{V}^{\star}\mathbf{\Lambda}_{\mathcal{L}}\right) \tag{B.41}$$

$$= \mathrm{Tr}\left(\mathbf{\Lambda}_{\mathcal{L}}\right), \tag{B.42}$$

which is maximum by selecting the largest d eigenvalues of $\mathbf{X}\mathbf{X}^{\mathsf{T}}$ in $\mathbf{\Lambda}_{\mathcal{L}}$. $\mathbf{X}\mathbf{X}^{\mathsf{T}}$ being symmetric, is admits the eigendecomposition $\mathbf{X}\mathbf{X}^{\mathsf{T}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathsf{T}}$, where $\mathbf{U} \in \mathbb{R}^{D\times D}$ contains the eigenvectors, and $\mathbf{\Lambda} \in \mathbb{R}^{D\times D}$ contains the ordered eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_D$ on its diagonal [1]. Finally, the solution of the PCA optimization problem reads as:

$$\mathbf{V}^{\star} = \mathbf{U}_d = \left[\mathbf{u}_1, \cdots, \mathbf{u}_d\right], \tag{B.43}$$

i.e. the collection of the first d eigenvectors of $\mathbf{X}\mathbf{X}^{\mathsf{T}}$. The orthonormality constraint for $\mathbf{V}^{\star}$, i.e. $\mathbf{V}^{\star^{\mathsf{T}}}\mathbf{V}^{\star} = \mathbf{Id}_d$, induced by the problem's constraints and the Lagrangian vanishing gradients wrt. $\mathbf{\Lambda}_{\mathcal{L}}$, is respected by definition of the eigenvectors.

■

1. Note that, as $\mathbf{X}\mathbf{X}^{\mathsf{T}}$ is positive-semidefinite, all of its eigenvalues are also strictly positive.

Appendix C

Chapter 3's appendix

C.1 Proof of Eq. (3.10)

Proof. The objective is to solve:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \mathbb{E}_{(x,y)\sim p_{x,y}}\left[\|y - f(x)\|_2^2\right], \\ \text{subject to} \quad & f \in \mathcal{H} \end{aligned} \tag{C.1}$$

Assuming that $f\colon \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y}$ is an Hilbert space equipped with the scalar product $\langle \cdot, \cdot \rangle_{\mathcal{Y}}$ and norm $\|\cdot\|_{\mathcal{Y}}$, it is proposed to solve the more generic problem:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \mathbb{E}_{(x,y)\sim p_{x,y}}\left[\|y - f(x)\|_{\mathcal{Y}}^2\right]. \\ \text{subject to} \quad & f \in \mathcal{H} \end{aligned} \tag{C.2}$$

Introducing the risk functional $\mathsf{R}(f) = \mathbb{E}_{(x,y)\sim p_{x,y}}\left[\|y - f(x)\|_{\mathcal{Y}}^2\right]$, it is possible to solve the problem described in Eq. (C.2) using the Fréchet derivation theory. The Fréchet derivative of $\mathsf{R}(f)$ is defined as:

$$D\mathsf{R}(f) = \lim_{\epsilon \to 0} \frac{\mathsf{R}(f + \epsilon h) - \mathsf{R}(f)}{\epsilon}, \tag{C.3}$$

where $h \in \mathcal{Y}^{\mathcal{X}}$. By definition, one has:

$$\mathsf{R}(f + \epsilon h) - \mathsf{R}(f) = \mathbb{E}_{(x,y)\sim p_{x,y}}\left[\|y - f(x) - \epsilon h(x)\|_{\mathcal{Y}}^2\right] - \mathbb{E}_{(x,y)\sim p_{x,y}}\left[\|y - f(x)\|_{\mathcal{Y}}^2\right] \tag{C.4}$$

$$= \mathbb{E}_{(x,y)\sim p_{x,y}}\left[\|y - f(x) - \epsilon h(x)\|_{\mathcal{Y}}^2 - \|y - f(x)\|_{\mathcal{Y}}^2\right]. \tag{C.5}$$

Noticing that:

$$\|y - f(x) - \epsilon h(x)\|_{\mathcal{Y}}^2 = \|y - f(x)\|_{\mathcal{Y}}^2 - 2\epsilon\langle y - f(x), h(x)\rangle_{\mathcal{Y}} + \epsilon^2 \|h(x)\|_{\mathcal{Y}}^2, \tag{C.6}$$

one obtains:

$$\mathsf{R}(f + \epsilon h) - \mathsf{R}(f) = \mathbb{E}_{(x,y)\sim p_{x,y}}\left[-2\epsilon\langle y - f(x), h(x)\rangle_{\mathcal{Y}} + \epsilon^2 \|h(x)\|_{\mathcal{Y}}^2\right] \tag{C.7}$$

$$= -2\epsilon\mathbb{E}_{(x,y)\sim p_{x,y}}\left[\langle y - f(x), h(x)\rangle_{\mathcal{Y}}\right] + \epsilon^2\mathbb{E}_{(x,y)\sim p_{x,y}}\left[\|h(x)\|_{\mathcal{Y}}^2\right]. \tag{C.8}$$

The Fréchet derivative of $\mathsf{R}(f)$ is thus expressed as:

$$D\mathsf{R}(f) = -2\mathbb{E}_{(x,y)\sim p_{x,y}}\left[\langle y - f(x), h(x)\rangle_{\mathcal{Y}}\right]. \tag{C.9}$$

A solution of the optimization problem presented in Eq. (C.2) is then given by:

$$\mathbb{E}_{(x,y)\sim p_{x,y}}\left[\langle y - f^{\star}(x), h(x)\rangle_{\mathcal{Y}}\right] = 0, \forall h, \tag{C.10}$$

which can be expressed as:

$$\mathbb{E}_{(x,y)\sim p_{x,y}}\left[\langle y - f^{\star}(x), h(x)\rangle_{\mathcal{Y}}\right] = \int_{\mathcal{X}}\int_{\mathcal{Y}}\langle y - f^{\star}(x), h(x)\rangle_{\mathcal{Y}} p_{x,y}(x,y)\,\mathrm{d}x\mathrm{d}y \tag{C.11}$$

$$\overset{(*)}{=} \int_{\mathcal{X}}\int_{\mathcal{Y}}\langle y - f^{\star}(x), h(x)\rangle_{\mathcal{Y}} p_{y|x}(y|x)\, p_x(x)\,\mathrm{d}x\mathrm{d}y \tag{C.12}$$

$$\overset{(**)}{=} \int_{\mathcal{X}}\left\langle \int_{\mathcal{Y}}(y - f^{\star}(x))\, p_{y|x}(y|x)\,\mathrm{d}y, h(x)\right\rangle_{\mathcal{Y}} p_x(x)\,\mathrm{d}x \tag{C.13}$$

$$= \int_{\mathcal{X}}\langle \mathbb{E}_{y|x\sim p_{y|x}}\left[y - f^{\star}(x)\,|x\right], h(x)\rangle_{\mathcal{Y}} p_x(x)\,\mathrm{d}x \tag{C.14}$$

$$= \mathbb{E}_{x\sim p_x}\left[\langle \mathbb{E}_{y|x\sim p_{y|x}}\left[y - f^{\star}(x)\,|x\right], h(x)\rangle_{\mathcal{Y}}\right]. \tag{C.15}$$

Note that $(*)$ is obtained from the definition of the conditional distribution $p_{y|x}(y|x) = p_{x,y}(x,y)/p_x(x)$, and that $(**)$ originates from the bilinearity of the scalar product. Specifically:

$$\int_{\mathcal{X}}\langle f(x), y\rangle_{\mathcal{Y}}\mathrm{d}x = \left\langle \int_{\mathcal{X}} f(x)\,\mathrm{d}x, y\right\rangle_{\mathcal{Y}}. \tag{C.16}$$

The Fréchet derivative vanishing condition, that must hold for any function h, becomes:

$$\mathbb{E}_{x\sim p_x}\left[\langle \mathbb{E}_{y|x\sim p_{y|x}}\left[y - f^{\star}(x)\,|x\right], h(x)\rangle_{\mathcal{Y}}\right] = 0, \forall h \tag{C.17}$$

$$\Leftrightarrow \int_{\mathcal{X}}\langle \mathbb{E}_{y|x\sim p_{y|x}}\left[y - f^{\star}(x)\,|x\right], h(x)\rangle_{\mathcal{Y}} p_x(x)\,\mathrm{d}x = 0, \forall h \tag{C.18}$$

$$\overset{(*)}{\Leftrightarrow} \mathbb{E}_{y|x\sim p_{y|x}}\left[y - f^{\star}(x)\,|x\right] = 0 \tag{C.19}$$

$$\Leftrightarrow f^{\star}(x) = \mathbb{E}_{y|x\sim p_{y|x}}\left[y|x\right], \tag{C.20}$$

which concludes the proof. Note that $(*)$ is obtained by remarking that, since the equality:

$$\int_{\mathcal{X}}\langle g(x), h(x)\rangle_{\mathcal{Y}} p_x(x)\,\mathrm{d}x = 0, \tag{C.21}$$

must hold for all h, it implies that $g(x) = 0$ almost everywhere (wrt. p_x). This is a direct manifestation of fundamental lemma of the calculus of variations. ■

Appendix D

Chapter 4's appendix

D.1 Proof of Lemma 4.1

Proof. Let $\mathbf{x}_r \in \mathbb{R}^3$ be a reference location and $\mathcal{V}_\mathbf{x} \subset \mathbb{R}^3$ be a local validity domain such that $\forall \mathbf{x} \in \mathcal{V}_\mathbf{x}, \|\mathbf{x} - \mathbf{x}_r\|_2 \leq \epsilon_\mathbf{x}$. The same holds true in the antenna subspace with the reference antenna location $\mathbf{a}_{l,r} \in \mathbb{R}^3$ and local validity domain $\mathcal{V}_\mathbf{a} \subset \mathbb{R}^3$ such that $\forall \mathbf{a}_{l,j} \in \mathcal{V}_\mathbf{a}, \|\mathbf{a}_{l,j} - \mathbf{a}_{l,r}\|_2 \leq \epsilon_\mathbf{a}$. Let $\psi(\mathbf{x}, \mathbf{a}_{l,j}) = \|\mathbf{x} - \mathbf{a}_{l,j}\|_2$, which is differentiable at $\mathbf{x} = \mathbf{x}_r$. The first order Taylor expansion of $\psi(\mathbf{x}, \mathbf{a}_{l,j})$ around $\mathbf{x}_r$ yields, $\forall \mathbf{x} \in \mathcal{V}_\mathbf{x}$:

$$\psi(\mathbf{x}, \mathbf{a}_{l,j}) \simeq \psi(\mathbf{x}_r, \mathbf{a}_{l,j}) + \nabla_\mathbf{x} \|\mathbf{x} - \mathbf{a}_{l,j}\|_2 \Big|_{\mathbf{x}_r}^{\mathsf{T}} (\mathbf{x} - \mathbf{x}_r) \tag{D.1}$$

$$= \|\mathbf{x}_r - \mathbf{a}_{l,j}\|_2 + \frac{(\mathbf{x}_r - \mathbf{a}_{l,j})^{\mathsf{T}}}{\|\mathbf{x}_r - \mathbf{a}_{l,j}\|_2} (\mathbf{x} - \mathbf{x}_r) \tag{D.2}$$

$$= \|\mathbf{x}_r - \mathbf{a}_{l,j}\|_2 + \mathbf{u}_{l,j} (\mathbf{x}_r)^{\mathsf{T}} (\mathbf{x} - \mathbf{x}_r). \tag{D.3}$$

In Eq. (D.3), one can remark the presence of $\psi(\mathbf{x}_r, \mathbf{a}_{l,j}) = \|\mathbf{x}_r - \mathbf{a}_{l,j}\|_2$, which is differentiable at $\mathbf{a}_{l,j} = \mathbf{a}_{l,r}$. As per the previous development, the first order Taylor expansion of $\psi(\mathbf{x}_r, \mathbf{a}_{l,j})$ around $\mathbf{a}_{l,r}$ yields, $\forall \mathbf{a}_{l,j} \in \mathcal{V}_\mathbf{a}$:

$$\psi(\mathbf{x}_r, \mathbf{a}_{l,j}) \simeq \|\mathbf{x}_r - \mathbf{a}_{l,r}\|_2 - \mathbf{u}_{l,r} (\mathbf{x}_r)^{\mathsf{T}} (\mathbf{a}_{l,j} - \mathbf{a}_{l,r}). \tag{D.4}$$

Finally one has, $\forall (\mathbf{x}, \mathbf{a}_{l,j}) \in \mathcal{V}_\mathbf{x} \times \mathcal{V}_\mathbf{a}$:

$$\|\mathbf{x} - \mathbf{a}_{l,j}\|_2 \simeq \|\mathbf{x}_r - \mathbf{a}_{l,r}\|_2 + \mathbf{u}_{l,j} (\mathbf{x}_r)^{\mathsf{T}} (\mathbf{x} - \mathbf{x}_r) - \mathbf{u}_{l,r} (\mathbf{x}_r)^{\mathsf{T}} (\mathbf{a}_{l,j} - \mathbf{a}_{l,r}), \tag{D.5}$$

which concludes the proof. ■

D.2 Proof of Corollary 4.1

Proof. Let $\mathbf{x}_r \in \mathbb{R}^3$ and $\mathbf{a}_{l,r} \in \mathbb{R}^3$ respectively denote a reference location and a reference antenna location. The approximation in Eq. (4.10) involves two Taylor expansions: a first one on $\psi_1(\mathbf{x}) = \|\mathbf{x} - \mathbf{a}_{l,j}\|_2$ around $\mathbf{x}_r$, and a second one on $\psi_2(\mathbf{a}_{l,j}) = \|\mathbf{x}_r - \mathbf{a}_{l,j}\|_2$ around $\mathbf{a}_{l,r}$. As such, the approximation error, denoted as e, can be approximated as the second order of these Taylor expansions, i.e.:

$$e \simeq e_1 + e_2, \tag{D.6}$$

where:

$$e_1 = \frac{1}{2}\left(\mathbf{x}-\mathbf{x}_r\right)^{\mathsf{T}} \nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathsf{T}}\psi_1\left(\mathbf{x}\right)\Big|_{\mathbf{x}_r}\left(\mathbf{x}-\mathbf{x}_r\right) \tag{D.7}$$

$$e_2 = \frac{1}{2}\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)^{\mathsf{T}} \nabla_{\mathbf{a}_{l,j}}\nabla_{\mathbf{a}_{l,j}}^{\mathsf{T}}\psi_2\left(\mathbf{a}_{l,j}\right)\Big|_{\mathbf{a}_{l,r}}\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right). \tag{D.8}$$

Remarking that $\psi_1\left(\mathbf{x}\right)$ is twice-differentiable:

$$\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathsf{T}}\left\|\mathbf{x}-\mathbf{a}_{l,j}\right\|_2 = \frac{\mathbf{Id}_3}{\left\|\mathbf{x}-\mathbf{a}_{l,j}\right\|_2} - \frac{\left(\mathbf{x}-\mathbf{a}_{l,j}\right)\left(\mathbf{x}-\mathbf{a}_{l,j}\right)^{\mathsf{T}}}{\left\|\mathbf{x}-\mathbf{a}_{l,j}\right\|_2^3}, \tag{D.9}$$

yields:

$$e_1 = \frac{1}{2}\left(\mathbf{x}-\mathbf{x}_r\right)^{\mathsf{T}}\left(\frac{\mathbf{Id}_3}{\left\|\mathbf{x}_r-\mathbf{a}_{l,j}\right\|_2} - \frac{\left(\mathbf{x}_r-\mathbf{a}_{l,j}\right)\left(\mathbf{x}_r-\mathbf{a}_{l,j}\right)^{\mathsf{T}}}{\left\|\mathbf{x}_r-\mathbf{a}_{l,j}\right\|_2^3}\right)\left(\mathbf{x}-\mathbf{x}_r\right) \tag{D.10}$$

$$\overset{(*)}{=} \frac{1}{2}\left(\frac{\left(\mathbf{x}-\mathbf{x}_r\right)^{\mathsf{T}}\left(\mathbf{x}-\mathbf{x}_r\right)}{\left\|\mathbf{x}_r-\mathbf{a}_{l,j}\right\|_2} - \frac{\left(\mathbf{x}-\mathbf{x}_r\right)^{\mathsf{T}}\mathbf{\Gamma}_{\mathbf{x}_r}\left(\mathbf{x}-\mathbf{x}_r\right)}{\left\|\mathbf{x}_r-\mathbf{a}_{l,j}\right\|_2^3}\right) \tag{D.11}$$

$$\overset{(**)}{=} \frac{1}{2}\left(\frac{\left\|\mathbf{x}-\mathbf{x}_r\right\|_2^2}{\left\|\mathbf{x}_r-\mathbf{a}_{l,j}\right\|_2} - o\left(\frac{1}{\left\|\mathbf{x}_r-\mathbf{a}_{l,j}\right\|_2^2}\right)\right), \tag{D.12}$$

where $(*)$ originates from $\mathbf{\Gamma}_{\mathbf{x}_r} = \left(\mathbf{x}_r-\mathbf{a}_{l,j}\right)\left(\mathbf{x}_r-\mathbf{a}_{l,j}\right)^{\mathsf{T}}$, and $(**)$ by considering:

$$\frac{\left(\mathbf{x}-\mathbf{x}_r\right)^{\mathsf{T}}\mathbf{\Gamma}_{\mathbf{x}_r}\left(\mathbf{x}-\mathbf{x}_r\right)}{\left\|\mathbf{x}_r-\mathbf{a}_{l,j}\right\|_2} \to 0. \tag{D.13}$$

In a similar manner, as $\psi_2\left(\mathbf{a}_{l,j}\right)$ is twice-differentiable:

$$\nabla_{\mathbf{a}_{l,j}}\nabla_{\mathbf{a}_{l,j}}^{\mathsf{T}}\left\|\mathbf{x}_r-\mathbf{a}_{l,j}\right\|_2 = \nabla_{\mathbf{a}_{l,j}}\nabla_{\mathbf{a}_{l,j}}^{\mathsf{T}}\left\|\mathbf{a}_{l,j}-\mathbf{x}_r\right\|_2 \tag{D.14}$$

$$= \frac{\mathbf{Id}_3}{\left\|\mathbf{a}_{l,j}-\mathbf{x}_r\right\|_2} - \frac{\left(\mathbf{a}_{l,j}-\mathbf{x}_r\right)\left(\mathbf{a}_{l,j}-\mathbf{x}_r\right)^{\mathsf{T}}}{\left\|\mathbf{a}_{l,j}-\mathbf{x}_r\right\|_2^3}, \tag{D.15}$$

yields:

$$e_2 = \frac{1}{2}\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)^{\mathsf{T}}\left(\frac{\mathbf{Id}_3}{\left\|\mathbf{a}_{l,r}-\mathbf{x}_r\right\|_2} - \frac{\left(\mathbf{a}_{l,r}-\mathbf{x}_r\right)\left(\mathbf{a}_{l,r}-\mathbf{x}_r\right)^{\mathsf{T}}}{\left\|\mathbf{a}_{l,r}-\mathbf{x}_r\right\|_2^3}\right)\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right) \tag{D.16}$$

$$\overset{(*)}{=} \frac{1}{2}\left(\frac{\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)^{\mathsf{T}}\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)}{\left\|\mathbf{a}_{l,r}-\mathbf{x}_r\right\|_2} - \frac{\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)^{\mathsf{T}}\mathbf{\Gamma}_{\mathbf{a}_r}\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)}{\left\|\mathbf{a}_{l,r}-\mathbf{x}_r\right\|_2^3}\right) \tag{D.17}$$

$$= \frac{1}{2}\left(\frac{\left\|\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right\|_2^2}{\left\|\mathbf{x}_r-\mathbf{a}_{l,r}\right\|_2} - \frac{\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)^{\mathsf{T}}\mathbf{\Gamma}_{\mathbf{a}_r}\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)}{\left\|\mathbf{x}_r-\mathbf{a}_{l,r}\right\|_2^3}\right) \tag{D.18}$$

$$\overset{(**)}{=} \frac{1}{2}\left(\frac{\left\|\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right\|_2^2}{\left\|\mathbf{x}_r-\mathbf{a}_{l,r}\right\|_2} - o\left(\frac{1}{\left\|\mathbf{x}_r-\mathbf{a}_{l,r}\right\|_2^2}\right)\right), \tag{D.19}$$

where $(*)$ originates from $\mathbf{\Gamma}_{\mathbf{a}_r} = (\mathbf{a}_{l,r}-\mathbf{x}_r)(\mathbf{a}_{l,r}-\mathbf{x}_r)^\mathsf{T}$, and $(**)$ by considering:

$$\frac{(\mathbf{a}_{l,j}-\mathbf{a}_{l,r})^\mathsf{T}\mathbf{\Gamma}_{\mathbf{a}_r}(\mathbf{a}_{l,j}-\mathbf{a}_{l,r})}{\left\|\mathbf{x}_r-\mathbf{a}_{l,r}\right\|_2} \to 0. \tag{D.20}$$

Finally, one obtains:

$$e \simeq \frac{1}{2}\left(\frac{\left\|\mathbf{x}-\mathbf{x}_r\right\|_2^2}{\left\|\mathbf{x}_r-\mathbf{a}_{l,j}\right\|_2} - o\left(\frac{1}{\left\|\mathbf{x}_r-\mathbf{a}_{l,j}\right\|_2^2}\right) + \frac{\left\|\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right\|_2^2}{\left\|\mathbf{x}_r-\mathbf{a}_{l,r}\right\|_2} - o\left(\frac{1}{\left\|\mathbf{x}_r-\mathbf{a}_{l,r}\right\|_2^2}\right)\right), \tag{D.21}$$

which concludes the proof. ■

D.3 Proof of Proposition 4.1

Proof. Let $f_r \in \mathbb{R}$ such that $\forall f_k \in \mathbb{R}, f_k = (f_k - f_r) + f_r$. Eq. (4.8) can be rewritten as:

$$h_{j,k}(\mathbf{x}) = \sum_{l=1}^{L_p} \frac{\gamma_l}{\left\|\mathbf{x}-\mathbf{a}_{l,j}\right\|_2} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\left\|\mathbf{x}-\mathbf{a}_{l,j}\right\|_2} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_{k-r}}\left\|\mathbf{x}-\mathbf{a}_{l,j}\right\|_2}, \tag{D.22}$$

where λ_r, resp. λ_{k-r}, is the wavelength associated to frequency f_r, resp. $f_k - f_r$. Let $\boldsymbol{\delta}_\mathbf{x} = \mathbf{x}-\mathbf{x}_r$ and $\boldsymbol{\delta}_\mathbf{a} = \mathbf{a}_{l,j}-\mathbf{a}_{l,r}$. Introducing Eq. (4.10) of Lemma 4.1 in Eq. (D.22) yields $\forall(\mathbf{x},\mathbf{a}_{l,j}) \in \mathcal{V}_\mathbf{x} \times \mathcal{V}_\mathbf{a}$:

$$h_{j,k}(\mathbf{x}) \simeq \sum_{l=1}^{L_p} \frac{\gamma_l}{\left\|\mathbf{x}-\mathbf{a}_{l,j}\right\|_2} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_{k-r}}\left\|\mathbf{x}_r-\mathbf{a}_{l,r}\right\|_2} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_{k-r}}\mathbf{u}_{l,j}(\mathbf{x}_r)^\mathsf{T}\boldsymbol{\delta}_\mathbf{x}} \mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_{k-r}}\mathbf{u}_{l,r}(\mathbf{x}_r)^\mathsf{T}\boldsymbol{\delta}_\mathbf{a}} \tag{D.23}$$

$$\cdot\, \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\left\|\mathbf{x}_r-\mathbf{a}_{l,r}\right\|_2} \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,j}(\mathbf{x}_r)^\mathsf{T}\boldsymbol{\delta}_\mathbf{x}} \mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^\mathsf{T}\boldsymbol{\delta}_\mathbf{a}}. \tag{D.24}$$

Considering that the reference frequency f_r is on the same order than the current frequency f_k gives:

$$f_k - f_r \ll f_r \Rightarrow \frac{\lambda_r}{\lambda_{k-r}} \ll 1. \tag{D.25}$$

By definition of the validity domains, one has $\forall \mathbf{a}_{l,j} \in \mathcal{V}_\mathbf{a}, \left\|\boldsymbol{\delta}_\mathbf{a}\right\|_2 \leq \epsilon_\mathbf{a}$ and $\forall \mathbf{x} \in \mathcal{V}_\mathbf{x}, \left\|\boldsymbol{\delta}_\mathbf{x}\right\|_2 \leq \epsilon_\mathbf{x}$. Thus, the location and antenna differences $\boldsymbol{\delta}_\mathbf{a}$ and $\boldsymbol{\delta}_\mathbf{x}$ can be expressed as a small number of the reference wavelength in both directions, i.e.:

$$\exists(\mathbf{m}_\mathbf{a},\mathbf{m}_\mathbf{x}) \in \mathbb{R}^3 \times \mathbb{R}^3 \text{ st. } \boldsymbol{\delta}_\mathbf{a} \simeq \mathbf{m}_\mathbf{a}\lambda_r, \boldsymbol{\delta}_\mathbf{x} \simeq \mathbf{m}_\mathbf{x}\lambda_r, \tag{D.26}$$

with $\|\mathbf{m_a}\|_2$ and $\|\mathbf{m_x}\|_2$ small. Then, $\forall\left(\mathbf{x},\mathbf{a}_{l,j}\right)\in\mathcal{V}_{\mathbf{x}}\times\mathcal{V}_{\mathbf{a}}$:

$$\begin{cases}\boldsymbol{\delta}_{\mathbf{a}}\simeq\mathbf{m_a}\lambda_r\\ \boldsymbol{\delta}_{\mathbf{x}}\simeq\mathbf{m_x}\lambda_r\end{cases}\Rightarrow\begin{cases}\left|\dfrac{\mathbf{u}_{l,r}\left(\mathbf{x}_r\right)^{\mathsf{T}}\boldsymbol{\delta}_{\mathbf{a}}}{\lambda_{k-r}}\right|\ll 1\\ \left|\dfrac{\mathbf{u}_{l,j}\left(\mathbf{x}_r\right)^{\mathsf{T}}\boldsymbol{\delta}_{\mathbf{x}}}{\lambda_{k-r}}\right|\ll 1\end{cases}. \tag{D.27}$$

Furthermore, as in classical communication systems the inter-antenna spacing is on the order of the wavelength, one can use the following approximation: $\forall\mathbf{x}_r\in\mathbb{R}^3,\mathbf{u}_{l,j}\left(\mathbf{x}_r\right)\simeq\mathbf{u}_{l,r}\left(\mathbf{x}_r\right)$. Namely that the DoD towards the reference location $\mathbf{x}_r$, for each antenna, can be approximated as the DoD of the reference antenna. The limitations of this approximation are discussed in this thesis. Introducing $d_{l,r}=\|\mathbf{x}_r-\mathbf{a}_{l,r}\|_2$, one then obtain, $\forall\left(\mathbf{x},\mathbf{a}_{l,j}\right)\in\mathcal{V}_{\mathbf{x}}\times\mathcal{V}_{\mathbf{a}}$:

$$h_{j,k}\left(\mathbf{x}\right)\simeq\sum_{l=1}^{L_p}\frac{\gamma_l}{\|\mathbf{x}-\mathbf{a}_{l,j}\|_2}\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}d_{l,r}}\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^{\mathsf{T}}\boldsymbol{\delta}_{\mathbf{x}}}\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_{k-r}}d_{l,r}}\mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^{\mathsf{T}}\boldsymbol{\delta}_{\mathbf{a}}}. \tag{D.28}$$

Additionally, as $\|\mathbf{x}-\mathbf{a}_{l,j}\|_2$ is a slowly varying term, one can approximate it as $d_{l,r}$ without suffering from a significant approximation error. One can then simplify the first two terms of Eq. (D.28) as $h_{l,r}\left(\mathbf{x}_r\right)=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}d_{l,r}}/d_{l,r}$. Finally, letting $\tau_{l,r}=d_{l,r}/c$ yields $2\pi d_{l,r}/\lambda_{k-r}=2\pi\left(f_k-f_r\right)\tau_{l,r}$. One finally obtains, $\forall\left(\mathbf{x},\mathbf{a}_{l,j}\right)\in\mathcal{V}_{\mathbf{x}}\times\mathcal{V}_{\mathbf{a}}$:

$$h_{j,k}\left(\mathbf{x}\right)\simeq\sum_{l=1}^{L_p}\gamma_l h_{l,r}\left(\mathbf{x}_r\right)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^{\mathsf{T}}(\mathbf{x}-\mathbf{x}_r)}\mathrm{e}^{-\mathrm{j}2\pi(f_k-f_r)\tau_{l,r}}\mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda_r}\mathbf{u}_{l,r}(\mathbf{x}_r)^{\mathsf{T}}\left(\mathbf{a}_{l,j}-\mathbf{a}_{l,r}\right)}, \tag{D.29}$$

which concludes the proof. ■

D.4 Proof of Theorem 4.2

Proof. Let us consider the tiling of the location subset $\mathcal{S}_{\mathbf{x}}\subset\mathbb{R}^3$ into $\Omega_{\mathbf{x}}$ location validity domains $\mathcal{V}_{\mathbf{x},i}$ with the Voronoi region of any given lattice. Note that for any subset of $\mathbb{R}^3$, the optimal lattice is the D_3-lattice. The same approach is applied to the antenna location subset $\mathcal{S}_{\mathbf{a}}\subset\mathbb{R}^3$, yielding $\Omega_{\mathbf{a}}$ local validity domains $\mathcal{V}_{\mathbf{a},i}$. By definition of the local validity domains and of the Taylor expansion, one has:

$$\forall\left(\mathcal{V}_{\mathbf{x},i},\mathcal{V}_{\mathbf{a},i}\right),\forall\left(\mathbf{x},\mathbf{a}_{l,j}\right)\in\mathcal{V}_{\mathbf{x},i}\times\mathcal{V}_{\mathbf{a},i},\left\|\mathbf{H}\left(\mathbf{x}\right)-\hat{\mathbf{H}}\left(\mathbf{x}\right)\right\|_{\mathsf{F}}<\epsilon, \tag{D.30}$$

where $\hat{\mathbf{H}}\left(\mathbf{x}\right)$ is the Taylor-approximated channel matrix computed using Eq. (4.14). For each local validity domain $\left(\mathcal{V}_{\mathbf{x},i},\mathcal{V}_{\mathbf{a},i}\right)$ pair, Eq. (4.14) shows that the channel can be approximated using only L_p planar wavefronts, SVs, and FRVs. Thus, one can construct a dictionary of planar wavefronts $\tilde{\boldsymbol{\psi}}_{\mathbf{x}}\in\mathbb{C}^D$, a dictionary of SVs $\tilde{\boldsymbol{\Psi}}_{\mathbf{a}}\in\mathbb{C}^{N_a\times D}$, and a dictionary of FRVs $\tilde{\boldsymbol{\Psi}}_{\mathbf{f}}\in\mathbb{C}^{N_s\times D}$ containing the needed atoms for every local validity

domain pair. This yields $D \leq L_p \Omega_{\mathbf{x}} \Omega_{\mathbf{a}}$.

While the FRV dictionary atoms follow Eq. (4.16), the SV dictionary atoms cannot be constituted from Eq. (4.15), as only the true antenna locations $\mathbf{a}_{1,j}$ are known. However, every virtual antenna can be rewritten as a rotated and translated version of its physical counterpart, i.e.:

$$\forall l \neq 1, \exists! \left(\mathbf{R}_{\theta,l}, \boldsymbol{\epsilon}_l\right) \in \mathrm{SO}\left(3\right) \times \mathbb{R}^3, \text{ s.t. } \mathbf{a}_{l,j} = \mathbf{R}_{\theta,l}\mathbf{a}_{1,j} + \boldsymbol{\epsilon}_l. \tag{D.31}$$

One then obtains:

$$\begin{aligned} \left(\mathbf{a}_{l,j} - \mathbf{a}_{l,r}\right) &= \mathbf{R}_{\theta,l}\mathbf{a}_{1,j} + \boldsymbol{\epsilon}_l - \mathbf{R}_{\theta,l}\mathbf{a}_{1,r} - \boldsymbol{\epsilon}_l & \text{(D.32)} \\ &= \mathbf{R}_{\theta,l}\left(\mathbf{a}_{1,j} - \mathbf{a}_{1,r}\right). & \text{(D.33)} \end{aligned}$$

While the antenna difference is not impacted by translations, it is impacted by rotations. However, one can easily show that the projection term in the SV is rotation equivariant:

$$\begin{aligned} \mathbf{u}_{l,r}\left(\mathbf{x}_r\right)^{\mathsf{T}}\left(\mathbf{a}_{l,j} - \mathbf{a}_{l,r}\right) &= \mathbf{u}_{l,r}\left(\mathbf{x}_r\right)^{\mathsf{T}}\mathbf{R}_{\theta,l}\left(\mathbf{a}_{1,j} - \mathbf{a}_{1,r}\right) & \text{(D.34)} \\ &= \left(\mathbf{R}_{\theta,l}^{\mathsf{T}}\mathbf{u}_{l,r}\left(\mathbf{x}_r\right)\right)^{\mathsf{T}}\left(\mathbf{a}_{1,j} - \mathbf{a}_{1,r}\right) & \text{(D.35)} \\ &= \tilde{\mathbf{u}}_{l,r}\left(\mathbf{x}_r\right)^{\mathsf{T}}\left(\mathbf{a}_{1,j} - \mathbf{a}_{1,r}\right). & \text{(D.36)} \end{aligned}$$

Thus, the dictionary of SVs can be constructed using DoDs $\tilde{\mathbf{u}}_i \in \mathbb{R}^3$ and the true antenna location difference $(\mathbf{a}_{1,j} - \mathbf{a}_{1,r})$. Using the previously defined dictionaries and introducing an activation vector $\mathbf{w}\left(\mathbf{x}\right) \in \mathbb{C}^D$ to select the needed L_p atoms at the current location $\mathbf{x}$, such that $\boldsymbol{\varpi}\left(\mathbf{x}\right) = \mathbf{w}\left(\mathbf{x}\right) \odot \tilde{\boldsymbol{\psi}}_{\mathbf{x}}\left(\mathbf{x}\right)$, yields, $\forall \mathbf{x} \in \mathbb{R}^3$:

$$\begin{aligned} &\mathbf{H}\left(\mathbf{x}\right) \simeq \sum_{i=1}^{D} \varpi_i\left(\mathbf{x}\right)\tilde{\boldsymbol{\psi}}_{\mathbf{a},i}\tilde{\boldsymbol{\psi}}_{\mathbf{f},i}^{\mathsf{T}}, & \text{(D.37)} \\ &\text{with } \left\|\boldsymbol{\varpi}\left(\mathbf{x}\right)\right\|_0 = L_p \end{aligned}$$

which concludes the proof. ■

Appendix E

Chapter 5's appendix

E.1 Proof of Theorem 5.1

Proof. From Definition 5.2, it follows that:

$$\forall \mathbf{x} \in \mathcal{S},\ \Xi(\mathbf{x}) \underset{\epsilon_{f_{\boldsymbol{\theta}}} \to 0}{\longrightarrow} 0 \tag{E.1}$$

$$\Rightarrow \forall \mathbf{x} \in \mathcal{S},\ f_{\boldsymbol{\theta}}(\mathbf{x}) \underset{\epsilon_{f_{\boldsymbol{\theta}}} \to 0}{\longrightarrow} \mathbf{H}(\mathbf{x}). \tag{E.2}$$

Assuming that Eq. (5.13) satisfies Definition 5.1, and invoking Definition 5.3, one obtains:

$$\forall \mathbf{x} \in \mathcal{S},\ \hat{\mathbf{x}}(\mathbf{H}(\mathbf{x}) | \boldsymbol{\theta}, \mathcal{S}) = \underset{\tilde{\mathbf{x}} \in \mathcal{S}}{\arg\min} \left\| \mathbf{H}(\mathbf{x}) - f_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) \right\|_{\mathsf{F}} \underset{\epsilon_{f_{\boldsymbol{\theta}}} \to 0}{\longrightarrow} \mathbf{x} \tag{E.3}$$

$$\Rightarrow \sup_{\mathbf{x} \in \mathcal{S}} \left\| \mathbf{x} - \hat{\mathbf{x}}(\mathbf{H}(\mathbf{x}) | \boldsymbol{\theta}, \mathcal{S}) \right\|_2 \underset{\epsilon_{f_{\boldsymbol{\theta}}} \to 0}{\longrightarrow} 0, \tag{E.4}$$

which concludes the proof. ■

E.2 Proof of Theorem 5.2

Proof. Let $\epsilon(\mathbf{x}) = \left\| \mathbf{x} - \hat{\mathbf{x}}(\mathbf{H}(\mathbf{x}) | \boldsymbol{\theta}, \mathcal{G}) \right\|_2$. A lower bound on $\mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}}[\epsilon(\mathbf{x})]$ can be derived by considering the best-case scenario, i.e. that the location estimate obtained by solving Eq. (5.12) on $\mathcal{G}$ is also optimal in the sense of the ℓ_2 norm, which reads as:

$$\hat{\mathbf{x}}(\mathbf{H}(\mathbf{x}) | \boldsymbol{\theta}, \mathcal{G}) = \underset{\tilde{\mathbf{x}} \in \mathcal{G}}{\arg\min} \left\| \mathbf{H}(\mathbf{x}) - f_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) \right\|_{\mathsf{F}} \tag{E.5}$$

$$= \underset{\tilde{\mathbf{x}} \in \mathcal{G}}{\arg\min} \left\| \mathbf{x} - \tilde{\mathbf{x}} \right\|_2. \tag{E.6}$$

By definition:

$$\mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}}[\epsilon(\mathbf{x})] = \int_{\mathbb{R}^2} \epsilon(\mathbf{x})\, p_{\mathbf{x}}(x, y)\, \mathrm{d}x\mathrm{d}y, \tag{E.7}$$

where $p_{\mathbf{x}}(x, y)$ is the probability density function of $p_{\mathbf{x}}$. Since under the considered assumptions, $\epsilon(\mathbf{x})$ denotes the distance between the true location and the closest point on the uniform grid $\mathcal{G}$ with inter-element spacing ν, it can be assumed that $\mathbf{x}$ is uniformly distributed within a square of side length ν and center $\mathbf{0}_{\mathbb{R}^2}$. However, due to the quadrant symmetries of this centered square, the computation of the overall expectation can be

reduced to that over a single quadrant. Consequently, $\mathbf{x}$ is uniformly distributed within a square of side length $\nu/2$, centered at $[\nu/4, \nu/4]^\mathsf{T}$, yielding:

$$p_{\mathbf{x}}(x, y) = \frac{4}{\nu^2} \mathbb{1}_{\{x \in \mathbb{R}^+ | x \leq \nu/2\}}(x) \, \mathbb{1}_{\{y \in \mathbb{R}^+ | y \leq \nu/2\}}(y) \,. \tag{E.8}$$

Note that, in this context, $\mathbf{x}_0 = [x_0, y_0]^\mathsf{T} = [\nu/2, \nu/2]^\mathsf{T}$ denotes the solution of Eq. (E.6), so that Eq. (E.7) can be expressed as:

$$\begin{aligned}
\mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}}[\epsilon(\mathbf{x})] &\geq \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}}[\|\mathbf{x} - \mathbf{x}_0\|_2] && \text{(E.9)}\\
&= \int_{\mathbb{R}^2} \sqrt{(x - x_0)^2 + (y - y_0)^2} p_{\mathbf{x}}(x, y) \, \mathrm{d}x\mathrm{d}y && \text{(E.10)}\\
&= \frac{4}{\nu^2} \int_0^{\nu/2} \int_0^{\nu/2} \sqrt{(x - x_0)^2 + (y - y_0)^2} \mathrm{d}x\mathrm{d}y && \text{(E.11)}\\
&\overset{(*)}{=} \frac{4}{\nu^2} \int_{-x_0}^{\nu/2 - x_0} \int_{-y_0}^{\nu/2 - y_0} \sqrt{x^2 + y^2} \mathrm{d}x\mathrm{d}y && \text{(E.12)}\\
&\overset{(**)}{=} \frac{4}{\nu^2} \int_{-\nu/2}^{0} \int_{-\nu/2}^{0} \sqrt{x^2 + y^2} \mathrm{d}x\mathrm{d}y && \text{(E.13)}\\
&\overset{(***)}{=} \frac{4}{\nu^2} \int_0^{\nu/2} \int_0^{\nu/2} \sqrt{x^2 + y^2} \mathrm{d}x\mathrm{d}y, && \text{(E.14)}
\end{aligned}$$

where $(*)$ originates from the change of variables $(x, y) \leftarrow (x - x_0, y - y_0)$, $(**)$ originates from the definition of $(x_0, y_0) = (\nu/2, \nu/2)$, and $(***)$ originates from the change of variables $(x, y) \leftarrow (-x, -y)$. Eq. (E.14) can then be rewritten in polar coordinates as:

$$\mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}}[\epsilon(\mathbf{x})] \geq \frac{4}{\nu^2} \int_0^{\frac{\pi}{2}} \int_0^{r_m(\theta)} r^2 \mathrm{d}\theta \mathrm{d}r, \tag{E.15}$$

where the axial integration bound is obtained from:

$$\begin{aligned}
\begin{cases} x = \frac{\nu}{2} \\ y = \frac{\nu}{2} \end{cases} \quad &\Rightarrow \begin{cases} r_m(\theta) = \frac{\nu}{2\cos\theta} \\ r_m(\theta) = \frac{\nu}{2\sin\theta} \end{cases} && \text{(E.16)}\\
&\Rightarrow r_m(\theta) = \min\left(\frac{\nu}{2\cos\theta}, \frac{\nu}{2\sin\theta}\right). && \text{(E.17)}
\end{aligned}$$

One then obtains:

$$\begin{aligned}
\mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}}[\epsilon(\mathbf{x})] &\geq \frac{4}{\nu^2} \left(\int_0^{\frac{\pi}{4}} \int_0^{\frac{\nu}{2\cos\theta}} r^2 \mathrm{d}\theta \mathrm{d}r + \int_{\frac{\pi}{4}}^{\frac{\pi}{2}} \int_0^{\frac{\nu}{2\sin\theta}} r^2 \mathrm{d}\theta \mathrm{d}r \right) && \text{(E.18)}\\
&= \frac{\nu}{6} \left(\int_0^{\frac{\pi}{4}} \frac{1}{\cos^3\theta} \mathrm{d}\theta + \int_{\frac{\pi}{4}}^{\frac{\pi}{2}} \frac{1}{\sin^3\theta} \mathrm{d}\theta \right). && \text{(E.19)}
\end{aligned}$$

Recalling that:

$$\int \frac{1}{\cos^3\theta}\mathrm{d}\theta = \frac{1}{2}\left(\sec\theta\tan\theta + \ln\left(|\sec\theta + \tan\theta|\right)\right) \tag{E.20}$$

$$\int \frac{1}{\sin^3\theta}\mathrm{d}\theta = \frac{1}{2}\left(-\csc\theta\cot\theta + \ln\left(|\csc\theta - \cot\theta|\right)\right), \tag{E.21}$$

yields the following lower bound:

$$\mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}}\left[\epsilon\left(\mathbf{x}\right)\right] \geq \frac{\nu}{6}\left(\sqrt{2} + \ln\left(\sqrt{2}+1\right)\right), \tag{E.22}$$

so that, $\forall \mathbf{x} \in \mathcal{S}$:

$$\mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}}\left[\left\|\mathbf{x} - \hat{\mathbf{x}}\left(\mathbf{H}\left(\mathbf{x}\right)|\boldsymbol{\theta}, \mathcal{G}\right)\right\|_2\right] \geq \nu\delta, \tag{E.23}$$

with $\delta = \frac{1}{6}\left(\sqrt{2} + \ln\left(\sqrt{2}+1\right)\right)$, which concludes the proof. ■

E.3 Proof of Theorem 5.3

Proof. From Eq. (5.8), introducing $d_{l,i}\left(\mathbf{x}\right) = \left\|\mathbf{x} - \mathbf{a}_{l,i}\right\|_2$ and $\phi_j = 2\pi/\lambda_j$ yields:

$$\left\|\mathbf{H}\left(\mathbf{x}\right) - \mathbf{H}\left(\mathbf{x}+\boldsymbol{\delta}\right)\right\|_{\mathsf{F}} = \sqrt{\sum_{i=1}^{N_a}\sum_{j=1}^{N_s}\left|\sum_{l=1}^{L_p}\gamma_l\left(\frac{\mathrm{e}^{-\mathrm{j}\phi_j d_{l,i}(\mathbf{x})}}{d_{l,i}\left(\mathbf{x}\right)} - \frac{\mathrm{e}^{-\mathrm{j}\phi_j d_{l,i}(\mathbf{x}+\boldsymbol{\delta})}}{d_{l,i}\left(\mathbf{x}+\boldsymbol{\delta}\right)}\right)\right|^2} \tag{E.24}$$

$$= \sqrt{\sum_{i=1}^{N_a}\sum_{j=1}^{N_s}\left|\sum_{l=1}^{L_p} f_{i,j,l}\left(\boldsymbol{\delta}\right)\right|^2}. \tag{E.25}$$

Considering $\left\|\boldsymbol{\delta}\right\|_2 \leq \epsilon_{\boldsymbol{\delta}}$, so that $d_{l,i}\left(\mathbf{x}\right)/d_{l,i}\left(\mathbf{x}+\boldsymbol{\delta}\right) \simeq 1$, $f_{i,j,l}\left(\boldsymbol{\delta}\right)$ can be rewritten as:

$$f_{i,j,l}\left(\boldsymbol{\delta}\right) \simeq \frac{\gamma_l}{d_{l,i}\left(\mathbf{x}\right)}\left(\mathrm{e}^{-\mathrm{j}\phi_j d_{l,i}(\mathbf{x})} - \mathrm{e}^{-\mathrm{j}\phi_j d_{l,i}(\mathbf{x}+\boldsymbol{\delta})}\right). \tag{E.26}$$

The function $\left\|\mathbf{H}\left(\mathbf{x}\right) - \mathbf{H}\left(\mathbf{x}+\boldsymbol{\delta}\right)\right\|_{\mathsf{F}}$ being lower-bounded by 0, a sufficient condition to obtain its global minimum is given by:

$$\boldsymbol{\delta} \in \bigcap_{(i,j,l)\in\mathcal{I}}\left\{\boldsymbol{\delta} \in \mathbb{R}^2 | f_{i,j,l}\left(\boldsymbol{\delta}\right) = 0\right\} \tag{E.27}$$

$$\Leftrightarrow \boldsymbol{\delta} \in \bigcap_{(i,j,l)\in\mathcal{I}}\left\{\boldsymbol{\delta} \in \mathbb{R}^2 | \mathrm{e}^{-\mathrm{j}\phi_j d_{l,i}(\mathbf{x})} = \mathrm{e}^{-\mathrm{j}\phi_j d_{l,i}(\mathbf{x}+\boldsymbol{\delta})}\right\}, \tag{E.28}$$

with $\mathcal{I} = [\![1, N_a]\!] \times [\![1, N_s]\!] \times [\![1, L_p]\!]$. Let $k \in \mathbb{Z}$, then:

$$\mathrm{e}^{-\mathrm{j}\phi_j d_{l,i}(\mathbf{x})} = \mathrm{e}^{-\mathrm{j}\phi_j d_{l,i}(\mathbf{x}+\boldsymbol{\delta})} \tag{E.29}$$

$$\Leftrightarrow \phi_j d_{l,i}\left(\mathbf{x}\right) = \phi_j d_{l,i}\left(\mathbf{x}+\boldsymbol{\delta}\right) + k2\pi \tag{E.30}$$

$$\Leftrightarrow \|\mathbf{x} - \mathbf{a}_{l,i} + \boldsymbol{\delta}\|_2 = \|\mathbf{x} - \mathbf{a}_{l,i}\|_2 + k\lambda_j \tag{E.31}$$

$$\Leftrightarrow \|\mathbf{x} + \boldsymbol{\delta} - \mathbf{a}_{l,i}\|_2 = \|\mathbf{x} - \mathbf{a}_{l,i}\|_2 + k\lambda_j \tag{E.32}$$

$$\overset{(*)}{\Leftrightarrow} \mathbf{x} + \boldsymbol{\delta} \in \mathcal{C}\left(\mathbf{a}_{l,i}, \|\mathbf{x} - \mathbf{a}_{l,i}\|_2 + k\lambda_j\right), \tag{E.33}$$

where $(*)$ originates from the definition of a circle. The global-minimum condition is thus given by:

$$\boldsymbol{\delta} \in \bigcap_{(i,j,l,k)\in\mathcal{L}} \mathcal{C}\left(\mathbf{a}_{l,i} - \mathbf{x}, \|\mathbf{x} - \mathbf{a}_{l,i}\|_2 + k\lambda_j\right), \tag{E.34}$$

with $\mathcal{L} = [\![1, N_a]\!] \times [\![1, N_s]\!] \times [\![1, L_p]\!] \times \mathbb{Z}$. Note that, in the absence of ambiguity, the circles intersect exclusively at $\mathbf{0}_{\mathbb{R}^2}$ for $k = 0$, implying that $\boldsymbol{\delta} = \mathbf{0}_{\mathbb{R}^2}$. When $\boldsymbol{\delta} = \mathbf{0}_{\mathbb{R}^2}$, $\|\mathbf{H}(\mathbf{x}) - \mathbf{H}(\mathbf{x} + \boldsymbol{\delta})\|_{\mathsf{F}}$ attains its minimum possible value, which confirms that Eq. (E.34) characterizes a global minimum. ■

E.4 Proof of Corollary 5.1

Proof. From Eq. (5.13), the similarity measure is maximized when the Frobenius distance is minimized. Hence, satisfying Definition 5.1 reduces to showing that $\boldsymbol{\delta} = \mathbf{0}_{\mathbb{R}^2}$ is the global minimum of $\|\mathbf{H}(\mathbf{x}) - \mathbf{H}(\mathbf{x} + \boldsymbol{\delta})\|_{\mathsf{F}}$, which was established in Theorem 5.3 under the no-ambiguity assumption. ■

E.5 Proof of Corollary 5.2

Proof. Assuming a dominant propagation path is equivalent to considering that:

$$\exists l_d \in [\![1, L_p]\!] \text{ s.t. } \forall l \in [\![1, L_p]\!], |\gamma_{l_d}| \gg |\gamma_l|. \tag{E.35}$$

This translates into negligible path components in Eq. (E.25) so that:

$$\|\mathbf{H}(\mathbf{x}) - \mathbf{H}(\mathbf{x} + \boldsymbol{\delta})\|_{\mathsf{F}} \simeq \sqrt{\sum_{i=1}^{N_a}\sum_{j=1}^{N_s}\left|\gamma_{l_d}\left(\frac{\mathrm{e}^{-\mathrm{j}\phi_j d_{l_d,i}(\mathbf{x})}}{d_{l_d,i}(\mathbf{x})} - \frac{\mathrm{e}^{-\mathrm{j}\phi_j d_{l_d,i}(\mathbf{x}+\boldsymbol{\delta})}}{d_{l_d,i}(\mathbf{x}+\boldsymbol{\delta})}\right)\right|^2}. \tag{E.36}$$

The minima condition defined in Eq. (5.18) can then be rewritten as:

$$\boldsymbol{\delta} \in \bigcap_{(i,j,k)\in\mathcal{J}} \mathcal{C}\left(\mathbf{a}_{l_d,i} - \mathbf{x}, \|\mathbf{x} - \mathbf{a}_{l_d,i}\|_2 + k\lambda_j\right), \tag{E.37}$$

with $\mathcal{J} = [\![1, N_a]\!] \times [\![1, N_s]\!] \times \mathbb{Z}$. The minima thus lie on circles centered at the locations of the antenna array elements, with radii determined by the system's wavelengths. In conventional communication systems, wavelength variation across the bandwidth is

negligible, such that $\forall j \in [\![1, N_s]\!]$:

$$\lambda_j \simeq \lambda_0 = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{c}{f_i}, \tag{E.38}$$

which yields:

$$\boldsymbol{\delta} \in \bigcap_{(i,k)\in\mathcal{K}} \mathcal{C}\left(\mathbf{a}_{l_d,i} - \mathbf{x}, \|\mathbf{x} - \mathbf{a}_{l_d,i}\|_2 + k\lambda_0\right), \tag{E.39}$$

with $\mathcal{K} = [\![1, N_a]\!] \times \mathbb{Z}$. In Eq. (E.39), the intersection of the circles originating from the different antennas depends only on the chosen integer $k \in \mathbb{Z}$, which translates into consecutive minima being approximately spaced by λ_0. ■

Appendix F

Chapter 7's appendix

F.1 About the subspace separation of $\mathbf{\Gamma_X}$

As presented in Eq. (7.1), in the considered system, the received signal expresses as:

$$\mathbf{X} = \mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\mathbf{S} + \mathbf{N}, \tag{F.1}$$

where $\mathbf{X} \in \mathbb{C}^{N\times T}$ corresponds to the measured signals, and $\boldsymbol{\theta} \in \left[-\pi/2, \pi/2\right]^{M}$ represents the sources DoAs, with $M < N$. The narrowband sources signals are defined as:

$$\mathbf{S} = \left[\mathbf{s}_1, \cdots, \mathbf{s}_T\right] \in \mathbb{C}^{M\times T}, \tag{F.2}$$

such that $\mathbf{s}_i \sim \mathcal{CN}\left(\mathbf{0}_M, \sigma_s^2\mathbf{Id}_M\right)$. The sensing noise $\mathbf{N} = \left[\mathbf{n}_1, \cdots, \mathbf{n}_T\right] \in \mathbb{C}^{N\times T}$ is defined with $\mathbf{n}_i \sim \mathcal{CN}\left(\mathbf{0}_N, \sigma_n^2\mathbf{Id}_N\right)$. $\mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right) = \left[\mathbf{a}_{\zeta}\left(\theta_1\right), \cdots, \mathbf{a}_{\zeta}\left(\theta_M\right)\right] \in \mathbb{C}^{N\times M}$ is a steering matrix. The covariance matrix of the received signal can be computed as:

$$\begin{aligned}
\mathbf{\Gamma_X} &= \mathbb{E}\left[\mathbf{X}\mathbf{X}^{\mathsf{H}}\right] && \text{(F.3)}\\
&= \mathbb{E}\left[\left(\mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\mathbf{S} + \mathbf{N}\right)\left(\mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\mathbf{S} + \mathbf{N}\right)^{\mathsf{H}}\right] && \text{(F.4)}\\
&= \mathbb{E}\left[\mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\mathbf{S}\mathbf{S}^{\mathsf{H}}\mathbf{A}_{\zeta}^{\mathsf{H}}\left(\boldsymbol{\theta}\right)\right] + \mathbb{E}\left[\mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\mathbf{S}\mathbf{N}^{\mathsf{H}}\right] + \mathbb{E}\left[\mathbf{N}\mathbf{S}^{\mathsf{H}}\mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\right] + \mathbb{E}\left[\mathbf{N}\mathbf{N}^{\mathsf{H}}\right] && \text{(F.5)}\\
&\overset{(*)}{=} \mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\mathbb{E}\left[\mathbf{S}\mathbf{S}^{\mathsf{H}}\right]\mathbf{A}_{\zeta}^{\mathsf{H}}\left(\boldsymbol{\theta}\right) + \mathbf{0} + \mathbf{0} + \mathbb{E}\left[\mathbf{N}\mathbf{N}^{\mathsf{H}}\right] && \text{(F.6)}\\
&= \mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\mathbf{\Gamma_S}\mathbf{A}_{\zeta}^{\mathsf{H}}\left(\boldsymbol{\theta}\right) + \sigma_n^2\mathbf{Id}_N, && \text{(F.7)}
\end{aligned}$$

where $(*)$ originates from the independence between $\mathbf{S}$ and $\mathbf{N}$, together with the fact that they represent centered random variables. Additionally, since uncorrelated sources are considered, one obtains $\mathbf{\Gamma_S} = \sigma_s^2\mathbf{Id}_M$, so that:

$$\mathbf{\Gamma_X} = \sigma_s^2\mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\mathbf{A}_{\zeta}^{\mathsf{H}}\left(\boldsymbol{\theta}\right) + \sigma_n^2\mathbf{Id}_N. \tag{F.8}$$

Assuming that the steering matrix is composed of independent columns yields:

$$\mathrm{rk}\left(\mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\right) = \mathrm{rk}\left(\mathbf{A}_{\zeta}^{\mathsf{H}}\left(\boldsymbol{\theta}\right)\right) = M, \tag{F.9}$$

so that:

$$\mathrm{rk}\left(\sigma_s^2\mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\mathbf{A}_{\zeta}^{\mathsf{H}}\left(\boldsymbol{\theta}\right)\right) \leq \min\left(\mathrm{rk}\left(\mathbf{A}_{\zeta}\left(\boldsymbol{\theta}\right)\right), \mathrm{rk}\left(\mathbf{A}_{\zeta}^{\mathsf{H}}\left(\boldsymbol{\theta}\right)\right)\right) \tag{F.10}$$

$$\leq \min(M, M) \tag{F.11}$$
$$= M. \tag{F.12}$$

Remark F.1. *This rank condition is not met when:*

$$\exists (i,j) \in [1, M]^2, i \neq j, \theta_i = \theta_j, \tag{F.13}$$

as in this case $\mathrm{rk}(\mathbf{A}_\zeta(\boldsymbol{\theta})) < M$*. Note that this implies to consider at least two-colocated sources. As such, this limit case can be ignored without loss of generality in the context of this demonstration.*

Now, remarking that $\mathbf{A}_\zeta(\boldsymbol{\theta})\mathbf{A}_\zeta^{\mathsf{H}}(\boldsymbol{\theta})$ is hermitian, it admits the following EVD:

$$\mathbf{A}_\zeta(\boldsymbol{\theta})\mathbf{A}_\zeta^{\mathsf{H}}(\boldsymbol{\theta}) = \mathbf{U}\boldsymbol{\Lambda}_\mathbf{A}\mathbf{U}^{\mathsf{H}}, \tag{F.14}$$

with $\boldsymbol{\Lambda}_\mathbf{A} \in \mathbb{C}^{N\times N}$ denoting a diagonal matrix holding the eigenvalues, and $\mathbf{U} \in \mathbb{C}^{N\times N}$ s.t. $\mathbf{U}^{\mathsf{H}}\mathbf{U} = \mathbf{U}\mathbf{U}^{\mathsf{H}} = \mathbf{Id}_N$ is the eigenvector matrix. Consequently, one can rewrite $\boldsymbol{\Gamma}_\mathbf{X}$ as follows:

$$\boldsymbol{\Gamma}_\mathbf{X} = \sigma_s^2\mathbf{A}_\zeta(\boldsymbol{\theta})\mathbf{A}_\zeta^{\mathsf{H}}(\boldsymbol{\theta}) + \sigma_n^2\mathbf{Id}_N \tag{F.15}$$
$$= \sigma_s^2\mathbf{A}_\zeta(\boldsymbol{\theta})\mathbf{A}_\zeta^{\mathsf{H}}(\boldsymbol{\theta}) + \sigma_n^2\mathbf{U}\mathbf{U}^{\mathsf{H}} \tag{F.16}$$
$$= \sigma_s^2\mathbf{U}\boldsymbol{\Lambda}_\mathbf{A}\mathbf{U}^{\mathsf{H}} + \sigma_n^2\mathbf{U}\mathbf{U}^{\mathsf{H}} \tag{F.17}$$
$$= \mathbf{U}\left(\sigma_s^2\boldsymbol{\Lambda}_\mathbf{A} + \sigma_n^2\mathbf{Id}_N\right)\mathbf{U}^{\mathsf{H}} \tag{F.18}$$
$$= \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathsf{H}}, \tag{F.19}$$

where $\boldsymbol{\Lambda} = \sigma_s^2\boldsymbol{\Lambda}_\mathbf{A} + \sigma_n^2\mathbf{Id}_N$. Eq. (F.19) corresponds to the EVD of $\boldsymbol{\Gamma}_\mathbf{X}$, and several comments can be made:

- As $\mathbf{A}_\zeta(\boldsymbol{\theta})\mathbf{A}_\zeta^{\mathsf{H}}(\boldsymbol{\theta}) \in \mathbb{C}^{N\times N}$, and $\mathrm{rk}\left(\mathbf{A}_\zeta(\boldsymbol{\theta})\mathbf{A}_\zeta^{\mathsf{H}}(\boldsymbol{\theta})\right) = M < N$, $\boldsymbol{\Lambda}_\mathbf{A}$ can be rewritten as a diagonal matrix with M non-null values, and $N - M$ null values, i.e.:

$$\boldsymbol{\Lambda}_\mathbf{A} = \mathrm{diag}\left(\tilde{\lambda}_1, \cdots, \tilde{\lambda}_M, 0, \cdots, 0\right). \tag{F.20}$$

- As a result, $\boldsymbol{\Lambda} = \sigma_s^2\boldsymbol{\Lambda}_\mathbf{A} + \sigma_n^2\mathbf{Id}_N$ admits the following structure:

$$\boldsymbol{\Lambda} = \mathrm{diag}\left(\sigma_s^2\tilde{\lambda}_1 + \sigma_n^2, \cdots, \sigma_s^2\tilde{\lambda}_M + \sigma_n^2, \sigma_n^2, \cdots, \sigma_n^2\right). \tag{F.21}$$

Consequently, the eigendecomposition of $\boldsymbol{\Gamma}_\mathbf{X}$ admits N eigenvalues with the following structure:

$$\lambda_i = \begin{cases} \sigma_s^2\tilde{\lambda}_i + \sigma_n^2, & i \in [\![1, M]\!] \\ \sigma_n^2, & i \in [\![M+1, N]\!] \end{cases}. \tag{F.22}$$

Additionally, from this structure, one can remark that the matrix containing the eigenvectors decomposes as:

$$\mathbf{U} = [\mathbf{U}_S, \mathbf{U}_N]\,, \tag{F.23}$$

where $\mathbf{U}_S \in \mathbb{C}^{N\times M}$, and $\mathbf{U}_N \in \mathbb{C}^{N\times(N-M)}$. Recalling that $\mathbf{U}^{\mathsf{H}}\mathbf{U} = \mathbf{Id}_N$ leads to:

$$\begin{aligned}
&\begin{bmatrix}\mathbf{U}_S & \mathbf{U}_N\end{bmatrix}^{\mathsf{H}}\begin{bmatrix}\mathbf{U}_S & \mathbf{U}_N\end{bmatrix} = \mathbf{Id}_N && \text{(F.24)}\\
\Leftrightarrow\ &\begin{bmatrix}\mathbf{U}_S^{\mathsf{H}}\\ \mathbf{U}_N^{\mathsf{H}}\end{bmatrix}\begin{bmatrix}\mathbf{U}_S & \mathbf{U}_N\end{bmatrix} = \mathbf{Id}_N && \text{(F.25)}\\
\Leftrightarrow\ &\begin{bmatrix}\mathbf{U}_S^{\mathsf{H}}\mathbf{U}_S & \mathbf{U}_S^{\mathsf{H}}\mathbf{U}_N\\ \mathbf{U}_N^{\mathsf{H}}\mathbf{U}_S & \mathbf{U}_N^{\mathsf{H}}\mathbf{U}_N\end{bmatrix} = \mathbf{Id}_N && \text{(F.26)}\\
\Leftrightarrow\ &\begin{cases}\mathbf{U}_S^{\mathsf{H}}\mathbf{U}_N = \mathbf{0}\\ \mathbf{U}_N^{\mathsf{H}}\mathbf{U}_S = \mathbf{0}\end{cases}, && \text{(F.27)}
\end{aligned}$$

which is equivalent to $\mathbf{U}_S \perp \mathbf{U}_N$. Note that the unitary constraint for $\mathbf{U}_S$ and $\mathbf{U}_N$ is verified by the definition of eigenvectors.

F.2 $\mu_{\mathrm{SL},P}$ and subspace orthogonalization

Let $\mathcal{U}_S \subset \mathbb{C}^N$ and $\mathcal{U}_N \subset \mathbb{C}^N$ respectively denote the signal and noise subspaces, such that:

$$\begin{aligned}
\dim_{\mathbb{R}}(\mathcal{U}_S) &= p && \text{(F.28)}\\
\dim_{\mathbb{R}}(\mathcal{U}_N) &= q && \text{(F.29)}\\
m &= \min(p,q)\,. && \text{(F.30)}
\end{aligned}$$

The principal angles between $\mathcal{U}_S$ and $\mathcal{U}_N$ are defined as [ZK13, Definition 2.1]:

$$\Theta(\mathcal{U}_S, \mathcal{U}_N) = [\theta_1, \cdots, \theta_m]\,, \tag{F.31}$$

where $\forall i, \theta_i \in [0, \pi/2]$. The principal angle θ_k is defined such that $\cos(\theta_k)$ is obtained by solving the following optimization problem:

$$\begin{aligned}
\cos(\theta_k) = \left|\mathbf{x}_k^{\mathsf{H}}\mathbf{y}_k\right| = \max_{\substack{\mathbf{x}\in\mathcal{U}_S,\\ \mathbf{y}\in\mathcal{U}_N}} \quad & \left|\mathbf{x}^{\mathsf{H}}\mathbf{y}\right|.\\
\text{s.t.} \quad & \left|\mathbf{x}^{\mathsf{H}}\mathbf{x}_i\right| = \left|\mathbf{y}^{\mathsf{H}}\mathbf{y}_i\right| = 0, \forall i < k,\\
& \|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = 1
\end{aligned}$$

Since this definition is not directly useful in its current form, it is proposed to exploit an alternative definition [ZK13, Theorem 2.1], that relies on $\mathbf{U}_S$ and $\mathbf{U}_N$, i.e. orthonormal

bases of $\mathcal{U}_S$ and $\mathcal{U}_N$:

$$\cos\left(\Theta\left(\mathcal{U}_S, \mathcal{U}_N\right)\right) = \sigma\left(\mathbf{U}_S^{\mathsf{H}}\mathbf{U}_N\right) = \sigma\left(\mathbf{U}_N^{\mathsf{H}}\mathbf{U}_S\right), \tag{F.32}$$

where $\sigma(\cdot)$ is the function mapping a matrix to its singular values.

In the context of the diffMUSIC method training presented in Section 7.3.2, one can consider the following principal-angle based loss function:

$$\mu_{\mathrm{SL},\Theta} = \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{\theta},\mathbf{X})\in\mathcal{D}}\sum_i \cos\left(\Theta\left(\mathcal{U}_S\left(\boldsymbol{\theta}|\boldsymbol{\zeta}\right), \mathcal{U}_N\left(\mathbf{X}\right)\right)_i\right), \tag{F.33}$$

$$= \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{\theta},\mathbf{X})\in\mathcal{D}}\sum_i \sigma_i\left(\mathbf{U}_N^{\mathsf{H}}\left(\mathbf{X}\right)\mathbf{U}_S\left(\boldsymbol{\theta}|\boldsymbol{\zeta}\right)\right), \tag{F.34}$$

where $\mathcal{U}_N(\mathbf{X})$ denotes the noise subspace computed from measurement $\mathbf{X}$, $\mathcal{U}_S(\boldsymbol{\theta}|\boldsymbol{\zeta})$ is the signal subspace associated with DoAs $\boldsymbol{\theta}$ and HWI $\boldsymbol{\zeta}$, and where $\mathbf{U}_N(\mathbf{X})$ and $\mathbf{U}_S(\boldsymbol{\theta}|\boldsymbol{\zeta})$ denote their respective orthonormal bases. Since two subspaces are orthogonal when their principal angles equal $\pi/2$, minimizing the sum of their cosines, as in Eq. (F.33), enforces orthogonality between the signal and noise subspaces, thereby producing sharp peaks at the correct locations in the MUSIC spectrum.

Considering the minimization of $\mu_{\mathrm{SL},P}$, defined in Eq. (7.21), expresses as:

$$\min \mu_{\mathrm{SL},P} = \min -\frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{\theta},\mathbf{X})\in\mathcal{D}}\sum_i P_{\mathrm{MUSIC}}\left(\theta_i|\boldsymbol{\zeta}\right) \tag{F.35}$$

$$= \max \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{\theta},\mathbf{X})\in\mathcal{D}}\sum_i \frac{1}{\left\|\mathbf{U}_N^{\mathsf{H}}\left(\mathbf{X}\right)\mathbf{a}_{\boldsymbol{\zeta}}\left(\theta_i\right)\right\|_2^2} \tag{F.36}$$

$$= \min \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{\theta},\mathbf{X})\in\mathcal{D}}\sum_i \left\|\mathbf{U}_N^{\mathsf{H}}\left(\mathbf{X}\right)\mathbf{a}_{\boldsymbol{\zeta}}\left(\theta_i\right)\right\|_2^2 \tag{F.37}$$

$$= \min \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{\theta},\mathbf{X})\in\mathcal{D}} \left\|\mathbf{U}_N^{\mathsf{H}}\left(\mathbf{X}\right)\mathbf{A}_{\boldsymbol{\zeta}}\left(\boldsymbol{\theta}\right)\right\|_{\mathsf{F}}^2 \tag{F.38}$$

$$\overset{(*)}{=} \min \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{\theta},\mathbf{X})\in\mathcal{D}}\sum_i \sigma_i^2\left(\mathbf{U}_N^{\mathsf{H}}\left(\mathbf{X}\right)\mathbf{A}_{\boldsymbol{\zeta}}\left(\boldsymbol{\theta}\right)\right), \tag{F.39}$$

where $(*)$ originates from $\|\mathbf{X}\|_{\mathsf{F}}^2 = \sum_i \sigma_i^2(\mathbf{X})$. Since $\mathrm{span}\left(\mathbf{A}_{\boldsymbol{\zeta}}(\boldsymbol{\theta})\right) = \mathrm{span}\left(\mathbf{U}_S(\boldsymbol{\theta}|\boldsymbol{\zeta})\right)$, and given that the correlation between two steering vectors with a DoA difference $\delta\theta$ becomes negligible when $\delta\theta$ exceeds the array's angular resolution, making them nearly orthogonal, $\mathbf{A}_{\boldsymbol{\zeta}}(\boldsymbol{\theta})$ can be viewed as an orthonormal basis of the signal subspace, assuming sufficiently spaced DoAs. This results in the following expression:

$$\min \mu_{\mathrm{SL},P} = \min \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{\theta},\mathbf{X})\in\mathcal{D}}\sum_i \sigma_i^2\left(\mathbf{U}_N^{\mathsf{H}}\left(\mathbf{X}\right)\mathbf{U}_S\left(\boldsymbol{\theta}|\boldsymbol{\zeta}\right)\right), \tag{F.40}$$

which establishes that minimizing $\mu_{\mathrm{SL},P}$ is equivalent to maximizing the distance between the noise and signal subspaces, as the left term in Eq. (F.40) is almost similar to the minimization of $\mu_{\mathrm{SL},\Theta}$, defined in Eq. (F.34), up to a square difference.

Bibliography

[3GP21] 3GPP, *New SI: Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface*, Meeting number 94e, e-Meeting, Dec. 2021.

[3GP22] 3GPP, *Final Summary of Evaluation on AI/ML for Positioning Accuracy Enhancement*, Meeting number 110bis-e, e-Meeting, Oct. 2022.

[3GP23] 3GPP, *5GNR, Physical layer procedures for data*, TS 38.214, V17.7.0, 2023.

[AA19] Muhammad Alrabeiah and Ahmed Alkhateeb, "Deep learning for TDD and FDD massive MIMO: mapping channels in space and frequency", *in*: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 1465–1470.

[Alk+18] Ahmed Alkhateeb, Sam Alex, Paul Varkey, Ying Li, Qi Qu, and Djordje Tujkovic, "Deep Learning Coordinated Beamforming for Highly-Mobile Millimeter Wave Systems", *in*: *IEEE Access* 6 (2018), pp. 37328–37348.

[Alk19] A. Alkhateeb, "DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications", *in*: *Proceedings of Information Theory and Applications Workshop*, San Diego, CA, 2019, pp. 1–8.

[Aze19] Chloé-Agathe Azencott, *Introduction au Machine Learning*, Dunod, 2019.

[Bar83] A. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms", *in*: *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 8, 1983, pp. 336–339.

[BD24] Emil Björnson and Özlem Tuğfe Demir, *Introduction to Multiple Antenna Communications and Reconfigurable Surfaces*, Now Publishers, 2024.

[BDA20] Eren Balevi, Akash Doshi, and Jeffrey G. Andrews, "Massive MIMO Channel Estimation With an Untrained Deep Neural Network", *in*: *IEEE Transactions on Wireless Communications* 19.*3* (2020), pp. 2079–2090.

[Bem+20] Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel, "X-fields: Implicit neural view-, light-and time-image interpolation", *in*: *ACM Transactions on Graphics* 39.*6* (2020), pp. 1–15.

[Ben75] Jon Louis Bentley, "Multidimensional binary search trees used for associative searching", *in*: *Communications of the ACM* 18.*9* (Sept. 1975), pp. 509–517.

[Bjö+13] Emil Björnson, Per Zetterberg, Mats Bengtsson, and Bjorn Ottersten, "Capacity Limits and Multiplexing Gains of MIMO Channels with Transceiver Impairments", *in*: *IEEE Communications Letters* 17.*1* (2013), pp. 91–94.

[Bjö+14] Emil Björnson, Jakob Hoydis, Marios Kountouris, and Mérouane Debbah, "Massive MIMO Systems With Non-Ideal Hardware: Energy Efficiency, Estimation, and Capacity Limits", *in*: *IEEE Transactions on Information Theory* 60.*11* (2014), pp. 7112–7139.

[Bjö+15] Emil Björnson, Michail Matthaiou, Antonios Pitarokoilis, and Erik G. Larsson, "Distributed massive MIMO in cellular networks: Impact of imperfect hardware and number of oscillators", *in*: *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 2436–2440.

[Boz+21] Farhad Bozorgi, Padmanava Sen, André N. Barreto, and Gerhard Fettweis, "RF Front-End Challenges for Joint Communication and Radar Sensing", *in*: *2021 1st IEEE International Online Symposium on Joint Communications & Sensing (JC&S)*, 2021, pp. 1–6.

[Bro+20] Tom Brown et al., "Language Models are Few-Shot Learners", *in*: *Advances in Neural Information Processing Systems*, ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901.

[BRY20] M. Majid Butt, Anil Rao, and Daejung Yoon, "RF Fingerprinting and Deep Learning Assisted UE Positioning in 5G", *in*: *IEEE Vehicular Technology Conference*, 2020, pp. 1–7.

[BSS19] Alexios Balatsoukas-Stimming and Christoph Studer, "Deep Unfolding for Communications Systems: A Survey and Some New Directions", *in*: *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*, 2019, pp. 266–271.

[BT96] Dimitri P. Bertsekas and John N. Tsitsiklis, *Neuro-dynamic programming.* Vol. 3, Athena Scientific, 1996.

[Bur+20] Daoud Burghal, Ashwin T. Ravi, Varun Rao, Abdullah A. Alghafis, and Andreas F. Molisch, "A Comprehensive Survey of Machine Learning Based Localization with Wireless Signals", *in*: *arXiv preprint* (2020).

[Cao+21] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu, "Towards Understanding the Spectral Bias of Deep Learning", *in*: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021, pp. 2205–2211.

[CC22a] Baptiste Chatelier and Matthieu Crussière, “On the Impact of Phase Noise on Beamforming Performance for mmWave Massive MIMO Systems”, *in*: *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2022, pp. 1563–1568.

[CC22b] Baptiste Chatelier and Matthieu Crussière, “On the Impact of Phase Noise on Beamforming Performance for mmWave Massive MIMO Systems”, *in*: *IEEE Wireless Communications and Networking Conference*, 2022, pp. 1563–1568.

[CD22] Mingyao Cui and Linglong Dai, “Channel Estimation for Extremely Large-Scale MIMO: Far-Field or Near-Field?”, *in*: *IEEE Transactions on Communications* 70.*4* (2022), pp. 2663–2677.

[CGS22] Yaodong Cui, Aihuang Guo, and Chunlin Song, “TransNet: Full Attention Network for CSI Feedback in FDD Massive MIMO System”, *in*: *IEEE Wireless Communications Letters* 11.*5* (2022), pp. 903–907.

[Cha+23] Baptiste Chatelier, Vincent Corlay, Cristina Ciochina, Fallou Coly, and Julien Guillet, “Influence of Dataset Parameters on the Performance of Direct UE Positioning via Deep Learning”, *in*: *2023 Joint European Conference on Networks and Communications&6G Summit (EuCNC/6G Summit)*, 2023, pp. 126–131.

[Cha+24a] Baptiste Chatelier, Vincent Corlay, Matthieu Crussiére, and Luc Le Magoarou, “CSI Compression Using Channel Charting”, *in*: *2024 58th Asilomar Conference on Signals, Systems, and Computers*, 2024, pp. 1660–1664.

[Cha+24b] Baptiste Chatelier, Luc Le Magoarou, Vincent Corlay, and Matthieu Crussière, “Model-Based Learning for Location-to-Channel Mapping”, *in*: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 12836–12840.

[Cha+25a] Baptiste Chatelier, Vincent Corlay, Matthieu Crussière, and Luc Le Magoarou, “Model-Based Learning for Multi-Antenna Multi-Frequency Location-to-Channel Mapping”, *in*: *IEEE Journal of Selected Topics in Signal Processing* (2025), pp. 1–16.

[Cha+25b] Baptiste Chatelier, Vincent Corlay, Musa Furkan Keskin, Matthieu Crussière, Henk Wymeersch, and Luc Le Magoarou, *Model-based Implicit Neural Representation for sub-wavelength Radio Localization*, 2025.

[Cha+25c] Baptiste Chatelier, José Miguel Mateos-Ramos, Vincent Corlay, Christian Häger, Matthieu Crussière, Henk Wymeersch, and Luc Le Magoarou, “Physically Parameterized Differentiable MUSIC for DoA Estimation with Uncalibrated Arrays”, *in*: *ICC 2025 - IEEE International Conference on Communications*, 2025, pp. 3858–3863.

[Che+22] Xiaohong Chen, Changxing Deng, Binggui Zhou, Huan Zhang, Guanghua Yang, and Shaodan Ma, "High-Accuracy CSI Feedback With Super-Resolution Network for Massive MIMO Systems", *in*: *IEEE Wireless Communications Letters* 11.*1* (2022), pp. 141–145.

[Che+24] Hui Chen, Musa Furkan Keskin, Sina Rezaei Aghdam, Hyowon Kim, Simon Lindberg, Andreas Wolfgang, Traian E. Abrudan, Thomas Eriksson, and Henk Wymeersch, "Modeling and Analysis of OFDM-Based 5G/6G Localization Under Hardware Impairments", *in*: *IEEE Transactions on Wireless Communications* 23.*7* (2024), pp. 7319–7333.

[Chn+22] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey, "Gaussian Activated Neural Radiance Fields for High Fidelity Reconstruction and Pose Estimation", *in*: *Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIII*, 2022, pp. 264–280.

[CLMR23] Baptiste Chatelier, Luc Le Magoarou, and Getachew Redieteab, "Efficient Deep Unfolding for SISO-OFDM Channel Estimation", *in*: *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 3450–3455.

[CLW21] Yinbo Chen, Sifei Liu, and Xiaolong Wang, "Learning continuous image representation with local implicit image function", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8628–8638.

[CNG24] Vincent Corlay, Viet-Hoa Nguyen, and Nicolas Gresset, "Probabilistic Positioning via Ray Tracing With Noisy Angle of Arrival Measurements", *in*: *IEEE Communications Letters* 28.*10* (2024), pp. 2278–2282.

[Col03] Sylvain Collonge, "Caractérisation et modélisation de la propagation des ondes électromagnétiques à 60 GHz à l'intérieur des bâtiments", PhD thesis, INSA de Rennes, Dec. 2003.

[Con+21] Jingyu Cong, Xianpeng Wang, Mengxing Huang, and Liangtian Wan, "Robust DoA Estimation Method for MIMO Radar via Deep Neural Networks", *in*: *IEEE Sensors Journal* 21.*6* (Mar. 2021), pp. 7498–7507.

[Cor+18] Vincent Corlay, Joseph J. Boutros, Philippe Ciblat, and Loïc Brunel, "Multilevel MIMO Detection with Deep Learning", *in*: *52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 1805–1809.

[Cor+22] Vincent Corlay, Joseph J. Boutros, Philippe Ciblat, and Loïc Brunel, "Neural Network Approaches to Point Lattice Decoding", *in*: *IEEE Transactions on Information Theory* 68.*5* (2022), pp. 2969–2989.

[CRT06] E.J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information", *in*: *IEEE Transactions on Information Theory* 52.*2* (2006), pp. 489–509.

[CSZ06] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, *Semi-Supervised Learning*, The MIT Press, Sept. 2006.

[Cuo+22] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli, "Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What"'s Next", *in*: *Journal of Scientific Computing* 92.*3* (July 2022).

[Cyb89] George Cybenko, "Approximation by superpositions of a sigmoidal function", *in*: *Mathematics of control, signals and systems* 2.*4* (1989), pp. 303–314.

[DE03] David L. Donoho and Michael Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization", *in*: *Proceedings of the National Academy of Sciences* 100.*5* (2003), pp. 2197–2202.

[Des19] Jeff Desjardins, *How much data is generated each day?*, https://www.weforum.org/stories/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/, World Economic Forum. Published in collaboration with Visual Capitalist, Apr. 2019.

[DiTar+14] Rocco Di Taranto, Srikar Muppirisetty, Ronald Raulefs, Dirk Slock, Tommy Svensson, and Henk Wymeersch, "Location-Aware Communications for 5G Networks: How location information can improve scalability, latency, and robustness of 5G", *in*: *IEEE Signal Processing Magazine* 31.*6* (2014), pp. 102–112.

[Don06] D.L. Donoho, "Compressed sensing", *in*: *IEEE Transactions on Information Theory* 52.*4* (2006), pp. 1289–1306.

[Du+21] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu, "Neural Radiance Flow for 4D View Synthesis and Video Processing", *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 14304–14314.

[ESB23] Florian Euchner, Phillip Stephan, and Stephan ten Brink, "Augmenting Channel Charting with Classical Wireless Source Localization Techniques", *in*: *2023 57th Asilomar Conference on Signals, Systems, and Computers*, 2023, pp. 1641–1647.

[Ess+25] Amer Essakine, Yanqi Cheng, Chun-Wun Cheng, Lipei Zhang, Zhongying Deng, Lei Zhu, Carola-Bibiane Schönlieb, and Angelica I Aviles-Rivero, "Where Do We Stand with Implicit Neural Representations? A Technical and Performance Survey", *in*: *Transactions on Machine Learning Research* (2025), Survey Certification.

[Far+11] Wissam Farjow, Abdellah Chehri, Mouftah Hussein, and Xavier Fernando, "Support Vector Machines for indoor sensor localization", *in*: *IEEE Wireless Communications and Networking Conference*, 2011, pp. 779–783.

[Fer+21] Paul Ferrand, Alexis Decurninge, Luis G. Ordoñez, and Maxime Guillaud, "Triplet-Based Wireless Channel Charting: Architecture and Experiments", *in*: *IEEE Journal on Selected Areas in Communications* 39.*8* (2021), pp. 2361–2373.

[Fle25] François Fleuret, *Tikz.net: manifold samples*, https://tikz.net/manifold//, 2025.

[Fon22] Corentin Fonteneau, "Communications à faible latence et à haute fiabilité dans les bandes millimétriques", PhD thesis, INSA de Rennes, June 2022.

[Fri19] Benjamin Friedlander, "Localization of Signals in the Near-Field of an Antenna Array", *in*: *IEEE Transactions on Signal Processing* 67.*15* (2019), pp. 3885–3893.

[Fuc04] J.-J. Fuchs, "On sparse representations in arbitrary redundant bases", *in*: *IEEE Transactions on Information Theory* 50.*6* (2004), pp. 1341–1344.

[FZ20] Zhiwei Fang and Justin Zhan, "Deep Physical Informed Neural Networks for Metamaterial Design", *in*: *IEEE Access* 8 (2020), pp. 24506–24513.

[Gao+18] Xuanxuan Gao, Shi Jin, Chao-Kai Wen, and Geoffrey Ye Li, "ComNet: Combination of Deep Learning and Expert Knowledge in OFDM Receivers", *in*: *IEEE Communications Letters* 22.*12* (2018), pp. 2627–2630.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, http://www.deeplearningbook.org, MIT Press, 2016.

[Ge+24] Yu Ge, Maximilian Stark, Musa Furkan Keskin, Frank Hofmann, Thomas Hansen, and Henk Wymeersch, "V2X Sidelink Positioning in FR1: Scenarios, Algorithms, and Performance Evaluation", *in*: *IEEE Journal on Selected Areas in Communications* 42.*10* (2024), pp. 2608–2624.

[GG05] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements", *in*: *IEEE Signal Processing Magazine* 22.*4* (2005), pp. 41–53.

[GL10] Karol Gregor and Yann LeCun, "Learning fast approximations of sparse coding", *in*: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, Haifa, Israel: Omnipress, 2010, pp. 399–406.

[GLMS25] Arad Gast, Luc Le Magoarou, and Nir Shlezinger, "DCD-MUSIC: Deep-Learning-Aided Cascaded Differentiable MUSIC Algorithm for Near-Field Localization of Multiple Sources", *in*: *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[Gon+22] Yubing Gong, Hanying Zhao, Keke Hu, Qianxi Lu, and Yuan Shen, "A Multipath-Aided Localization Method for MIMO-OFDM Systems via Tensor Decomposition", *in*: *IEEE Wireless Communications Letters* 11.*6* (2022), pp. 1225–1228.

[Gon+24] Tierui Gong, Li Wei, Chongwen Huang, George C. Alexandropoulos, Mérouane Debbah, and Chau Yuen, "Near-Field Channel Modeling for Holographic Mimo Communications", *in*: *IEEE Wireless Communications* 31.*3* (2024), pp. 108–116.

[Gul+20] Anmol Gulati et al., "Conformer: Convolution-augmented Transformer for Speech Recognition", *in*: *Interspeech 2020*, Oct. 2020.

[Guo+20] Jiajia Guo, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li, "Convolutional Neural Network-Based Multiple-Rate Compressive Sensing for Massive MIMO CSI Feedback: Design, Simulation, and Analysis", *in*: *IEEE Transactions on Wireless Communications* 19.*4* (2020), pp. 2827–2840.

[Guo+22] Jiajia Guo, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li, "Overview of Deep Learning-Based CSI Feedback in Massive MIMO Systems", *in*: *IEEE Transactions on Communications* 70.*12* (2022), pp. 8017–8045.

[GWJ21] Jiajia Guo, Chao-Kai Wen, and Shi Jin, "Deep Learning-Based CSI Feedback for Beamforming in Single- and Multi-Cell Massive MIMO Systems", *in*: *IEEE Journal on Selected Areas in Communications* 39.*7* (2021), pp. 1872–1884.

[HB91] Y.-D. Huang and M. Barkat, "Near-field multiple source localization by passive sensor array", *in*: *IEEE Transactions on Antennas and Propagation* 39.*7* (1991), pp. 968–975.

[HDL17] David Ha, Andrew M. Dai, and Quoc V. Le, "HyperNetworks", *in*: *International Conference on Learning Representations*, 2017.

[He+18] Hengtao He, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li, "Deep Learning-Based Channel Estimation for Beamspace mmWave Massive MIMO Systems", *in*: *IEEE Wireless Communications Letters* 7.*5* (2018), pp. 852–855.

[Hoa+18] Minh Tu Hoang, Yizhou Zhu, Brosnan Yuen, Tyler Reese, Xiaodai Dong, Tao Lu, Robert Westendorp, and Michael Xie, "A Soft Range Limited K-Nearest Neighbors Algorithm for Indoor Localization Enhancement", *in*: *IEEE Sensors Journal* 18.*24* (2018), pp. 10208–10216.

[Hoy+22] Jakob Hoydis, Sebastian Cammerer, Fayçal Ait Aoudia, Avinash Vem, Nikolaus Binder, Guillermo Marcus, and Alexander Keller, "Sionna: An Open-Source Library for Next-Generation Physical Layer Research", *in*: *arXiv preprint* (2022).

[Hoy+24] Jakob Hoydis, Fayçal Aït Aoudia, Sebastian Cammerer, Florian Euchner, Merlin Nimier-David, Stephan Ten Brink, and Alexander Keller, "Learning Radio Environments by Differentiable Ray Tracing", *in*: *IEEE Transactions on Machine Learning in Communications and Networking* 2 (2024), pp. 1527–1539.

[HR10] Thomas Höhne and Ville Ranki, "Phase Noise in Beamforming", *in*: *IEEE Transactions on Wireless Communications* 9.*12* (2010), pp. 3682–3689.

[HRW14] John R. Hershey, Jonathan Le Roux, and Felix Weninger, *Deep Unfolding: Model-Based Inspiration of Novel Deep Architectures*, 2014.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White, "Multilayer feedforward networks are universal approximators", *in*: *Neural networks* 2.*5* (1989), pp. 359–366.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The elements of statistical learning: data mining, inference and prediction*, Second, Springer, 2009.

[JGH18] Arthur Jacot, Franck Gabriel, and Clement Hongler, "Neural Tangent Kernel: Convergence and Generalization in Neural Networks", *in*: *Advances in Neural Information Processing Systems*, ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, vol. 31, Curran Associates, Inc., 2018.

[Ji+24] Junkai Ji, Wei Mao, Feng Xi, and Shengyao Chen, "TransMUSIC: A Transformer-Aided Subspace Method for DoA Estimation with Low-Resolution ADCS", *in*: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Apr. 2024, pp. 8576–8580.

[JohXu+20] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma, "Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks", *in*: *Communications in Computational Physics* 28.*5* (2020), pp. 1746–1767.

[Kaz+23] Parham Kazemi, Hanan Al-Tous, Tushara Ponnada, Christoph Studer, and Olav Tirkkonen, "Beam SNR Prediction Using Channel Charting", *in*: *IEEE Transactions on Vehicular Technology* 72.*10* (2023), pp. 13130–13145.

[KB15] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization", *in*: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ed. by Yoshua Bengio and Yann LeCun, 2015.

[KGW09] Mohammad A. Khojastepour, Krishna Gomadam, and Xiaodong Wang, "Pilot-assisted channel estimation for MIMO OFDM systems using theory of sparse signal recovery", *in*: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 2693–2696.

[KI08] Taehyuk Kang and Ronald A. Iltis, "Matching pursuits channel estimation for an underwater acoustic OFDM modem", *in*: *2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 5296–5299.

[Kim+98] Dong Kyu Kim, Sang Hyun Do, Hong Bae Cho, Hyung Jin Chol, and Ki Bum Kim, "A new joint algorithm of symbol timing recovery and sampling clock adjustment for OFDM systems", *in*: *IEEE Transactions on Consumer Electronics* 44.*3* (1998), pp. 1142–1149.

[Kla+25] Nay Klaimi, Amira Bedoui, Clément Elvira, Philippe Mary, and Luc Le Magoarou, "Model-Based Learning for Joint Channel Estimation and Hybrid MIMO Precoding", *in*: *2025 IEEE 26th International Workshop on Signal Processing and Artificial Intelligence for Wireless Communications (SPAWC)*, 2025, pp. 1–5.

[Kru64] J. B. Kruskal, "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis", *in*: *Psychometrika* 29.*1* (1964), pp. 1–27.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *in*: *Advances in Neural Information Processing Systems*, ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, vol. 25, Curran Associates, Inc., 2012.

[KY04] G. Z. Karabulut and A. Yongacoglu, "Sparse channel estimation using orthogonal matching pursuit algorithm", *in*: *IEEE 60th IEEE Vehicular Technology Magazine (VTC Fall)*, vol. 6, 2004, 3880–3884 Vol. 6.

[LD23] Yu Lu and Linglong Dai, "Near-Field Channel Estimation in Mixed LoS/NLoS Environments for Extremely Large-Scale MIMO Systems", *in*: *IEEE Transactions on Communications* 71.*6* (2023), pp. 3694–3707.

[LeC+89] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard, and Lawrence Jackel, "Handwritten Digit Recognition with a Back-Propagation Network", *in*: *Advances in Neural Information Processing Systems*, ed. by D. Touretzky, vol. 2, Morgan-Kaufmann, 1989.

[Lei+19] Eric Lei, Oscar Castañeda, Olav Tirkkonen, Tom Goldstein, and Christoph Studer, "Siamese Neural Networks for Wireless Positioning and Channel Charting", *in*: *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 200–207.

[LeMag+22a] Luc Le Magoarou, Taha Yassine, Stéphane Paquelet, and Matthieu Crussière, "Channel charting based beamforming", *in*: *2022 56th Asilomar Conference on Signals, Systems, and Computers*, 2022, pp. 1185–1189.

[LeMag+22b] Luc Le Magoarou, Taha Yassine, Stéphane Paquelet, and Matthieu Crussière, "Deep Learning for Location Based Beamforming with Nlos Channels", *in*: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8812–8816.

[LeMag21a] Luc Le Magoarou, "Efficient Channel Charting via Phase-Insensitive Distance Computation", *in*: *IEEE Wireless Communications Letters* 10.*12* (2021), pp. 2634–2638.

[LeMag21b] Luc Le Magoarou, "Similarity-Based Prediction for Channel Mapping and User Positioning", *in*: *IEEE Communications Letters* 25 (5 2021), pp. 1578–1582.

[Liu+07] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu, "Survey of Wireless Indoor Positioning Techniques and Systems", *in*: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.*6* (2007), pp. 1067–1080.

[Liu+14] Deliang Liu, Kaihua Liu, Yongtao Ma, and Jiexiao Yu, "Joint TOA and DOA Localization in Indoor Environment Using Virtual Stations", *in*: *IEEE Communications Letters* 18.*8* (2014), pp. 1423–1426.

[Liu+25] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark, "KAN: Kolmogorov–Arnold Networks", *in*: *The Thirteenth International Conference on Learning Representations*, 2025.

[LMLCP19] Luc Le Magoarou, Antoine Le Calvez, and Stéphane Paquelet, "Massive MIMO Channel Estimation taking into account spherical waves", *in*: *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019, pp. 1–5.

[LS23] Ortal Lavi and Nir Shlezinger, "Learn to Rapidly and Robustly Optimize Hybrid Precoding", *in*: *IEEE Transactions on Communications* 71.*10* (2023), pp. 5814–5830.

[Lu+17] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang, "The expressive power of neural networks: a view from the width", *in*: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6232–6240.

[Lu+21] Lu Lu, Raphaël Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G. Johnson, "Physics-Informed Neural Networks with Hard Constraints for Inverse Design", *in*: *SIAM Journal on Scientific Computing* 43.*6* (2021), B1105–B1132.

[LWS20] Zhilin Lu, Jintao Wang, and Jian Song, "Multi-resolution CSI Feedback with Deep Learning in Massive MIMO System", *in*: *ICC 2020 - 2020 IEEE International Conference on Communications*, 2020, pp. 1–6.

[LZW23] Xiaoyu Lan, Huichao Zhai, and Yupeng Wang, "A Novel DoA Estimation of Closely Spaced Sources Using Attention Mechanism With Conformal Arrays", *in*: *IEEE Access* 11 (2023), pp. 44010–44018.

[Mac67] J. B. MacQueen, "Some Methods for Classification and Analysis of MultiVariate Observations", *in*: *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, ed. by L. M. Le Cam and J. Neyman, vol. 1, University of California Press, 1967, pp. 281–297.

[MatRam+23] José Miguel Mateos-Ramos, Christian Häger, Musa Furkan Keskin, Luc Le Magoarou, and Henk Wymeersch, "Model-Driven End-to-End Learning for Integrated Sensing and Communication", *in*: *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 5695–5700.

[MatRam+24] José Miguel Mateos-Ramos, Baptiste Chatelier, Christian Häger, Musa Furkan Keskin, Luc Le Magoarou, and Henk Wymeersch, "Semi-Supervised End-to-End Learning for Integrated Sensing and Communications", *in*: *2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, 2024, pp. 132–138.

[MatRam+25a] José Miguel Mateos-Ramos, Christian Häger, Musa Furkan Keskin, Luc Le Magoarou, and Henk Wymeersch, "Model-Based End-to-End Learning for Multi-Target Integrated Sensing and Communication Under Hardware Impairments", *in*: *IEEE Transactions on Wireless Communications* 24.*3* (2025), pp. 2574–2589.

[MatRam+25b] José Miguel Mateos-Ramos, Christian Häger, Musa Furkan Keskin, Luc Le Magoarou, and Henk Wymeersch, "Unsupervised Learning for Gain-Phase Impairment Calibration in ISAC Systems", *in*: *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[MEE10] Roy Maiberger, Doron Ezri, and Michael Erlihson, "Location based beamforming", *in*: *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, IEEE, 2010.

[Mil+21] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis", *in*: *Communications of the ACM* 65.*1* (2021), pp. 99–106.

[MLE21] Vishal Monga, Yuelong Li, and Yonina C. Eldar, "Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing", *in*: *IEEE Signal Processing Magazine* 38.*2* (2021), pp. 18–44.

[MP43] Warren S. McCulloch and Walter Pitts, "A logical calculus of the ideas immanent in nervous activity", *in*: *The bulletin of mathematical biophysics* 5.*4* (Dec. 1943), pp. 115–133.

[MZ93] S.G. Mallat and Zhifeng Zhang, "Matching pursuits with time-frequency dictionaries", *in*: *IEEE Transactions on Signal Processing* 41.*12* (1993), pp. 3397–3415.

[Nat95] B. K. Natarajan, "Sparse Approximate Solutions to Linear Systems", *in*: *SIAM Journal on Computing* 24.*2* (1995), pp. 227–234.

[Nes+20] Ahasanun Nessa, Bhagawat Adhikari, Fatima Hussain, and Xavier N. Fernando, "A Survey of Machine Learning for Indoor Positioning", *in*: *IEEE Access* 8 (2020), pp. 214945–214965.

[Neu25a] Izaak Neutelings, *Tikz.net: diffraction samples*, https://tikz.net/optics_diffraction/, 2025.

[Neu25b] Izaak Neutelings, *Tikz.net: neural networks*, https://tikz.net/neural_networks/, 2025.

[Neu25c] Izaak Neutelings, *Tikz.net: refractions samples*, https://tikz.net/optics_refractions/, 2025.

[Nov62] Albert B. J. Novikoff, "On Convergence Proofs on Perceptrons", *in*: *Proceedings of the Symposium on the Mathematical Theory of Automata* 12 (1962), pp. 615–622.

[OH17] Timothy O'Shea and Jakob Hoydis, "An Introduction to Deep Learning for the Physical Layer", *in*: *IEEE Transactions on Cognitive Communications and Networking* 3.*4* (2017), pp. 563–575.

[Omo89] Stepphen M. Omohundro, *Five Balltree Construction Algorithms*, tech. rep., International Computer Science Institute, 1989.

[Par+21] Sejun Park, Chulhee Yun, Jaeho Lee, and Jinwoo Shin, "Minimum Width for Universal Approximation", *in*: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[Par62] Emanuel Parzen, "On Estimation of a Probability Density Function and Mode", *in*: *The Annals of Mathematical Statistics* 33.*3* (Sept. 1962), pp. 1065–1076.

[PDL25] Bumsu Park, Heedong Do, and Namyoon Lee, *Transformer-Based Nonlinear Transform Coding for Multi-Rate CSI Compression in MIMO-OFDM Systems*, 2025.

[Pea01] Karl Pearson, "On lines and planes of closest fit to systems of points in space", *in*: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.*11* (Nov. 1901), pp. 559–572.

[PGPR22] Joan Palacios, Nuria González-Prelcic, and Cristian Rusu, *Multidimensional orthogonal matching pursuit: theory and application to high accuracy joint localization and communication at mmWave*, 2022.

[Pon+21] Tushara Ponnada, Parham Kazemi, Hanan Al-Tous, Ying-Chang Liang, and Olav Tirkkonen, "Best Beam Prediction in Non-Standalone mmWave Systems", *in*: *2021 Joint European Conference on Networks and Communications&6G Summit (EuCNC/6G Summit)*, IEEE, 2021.

[Poz98] David M Pozar, *Microwave engineering*, Second, John Wiley & Sons, 1998.

[PRK93] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition", *in*: *1993 27th Asilomar Conference on Signals, Systems, and Computers*, 1993, 40–44 vol.1.

[PSE21] Georgios Papageorgiou, Mathini Sellathurai, and Yonina Eldar, "Deep Networks for Direction-of-Arrival Estimation in Low SNR", *in*: *IEEE Transactions on Signal Processing* 69 (2021), pp. 3714–3729.

[PTHV23] Marius Pesavento, Minh Trinh-Hoang, and Mats Viberg, "Three More Decades in Array Signal Processing Research: An optimization and structure exploitation perspective", *in*: *IEEE Signal Processing Magazine* 40.*4* (2023), pp. 92–106.

[Pum+21] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer, "D-NeRF: Neural Radiance Fields for Dynamic Scenes", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10313–10322.

[PX23] William Peebles and Saining Xie, "Scalable Diffusion Models with Transformers", *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 4195–4205.

[Qin+20] Yiwen Qin, Zhihong Feng, Nan Su, and Ding Ma, "Interior Fusion Localization Method Based on Random Forest", *in*: *International Conference on Measuring Technology and Mechatronics Automation*, 2020, pp. 371–374.

[Rah+19] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville, "On the spectral bias of neural networks", *in*: *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 5301–5310.

[Rak12] Sasha Rakhlin, *FStatistical Learning Theory*, 2012.

[Ras99] Carl Rasmussen, "The Infinite Gaussian Mixture Model", *in*: *Advances in Neural Information Processing Systems*, ed. by S. Solla, T. Leen, and K. Müller, vol. 12, MIT Press, 1999.

[Rev+21] Guy Revach, Nir Shlezinger, Ruud J. G. van Sloun, and Yonina C. Eldar, "Kalmannet: Data-Driven Kalman Filtering", *in*: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3905–3909.

[Rev+22] Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adrià López Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar, "KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics", *in*: *IEEE Transactions on Signal Processing* 70 (2022), pp. 1532–1547.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, "Learning representations by back-propagating errors", *in*: *Nature* 323.*6088* (Oct. 1986), pp. 533–536.

[RK89] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques", *in*: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.*7* (July 1989), pp. 984–995.

[RL22] Sameera Ramasinghe and Simon Lucey, "Beyond Periodicity: Towards a Unifying Framework for Activations in Coordinate-MLPs", *in*: *Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIII*, Tel Aviv, Israel: Springer-Verlag, 2022, pp. 142–158.

[RML22] Sameera Ramasinghe, Lachlan E. MacDonald, and Simon Lucey, "On the Frequency-bias of Coordinate-MLPs", *in*: *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 796–809.

[Roh+23] Franz M. Rohrhofer, Stefan Posch, Clemens Gößnitzer, and Bernhard C. Geiger, "Data vs. Physics: The Apparent Pareto Front of Physics-Informed Neural Networks", *in*: *IEEE Access* 11 (2023), pp. 86252–86261.

[Ros56] Murray Rosenblatt, "Remarks on Some Nonparametric Estimates of a Density Function", *in*: *The Annals of Mathematical Statistics* 27.*3* (Sept. 1956), pp. 832–837.

[Ros57] Frank Rosenblatt, *The perceptron: A perceiving and recognizing automaton*, Report, Project PARA, Cornell Aeronautical Laboratory, Jan. 1957.

[RPK19] M. Raissi, P. Perdikaris, and G.E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", *in*: *Journal of Computational Physics* 378 (2019), pp. 686–707.

[RR07] Ali Rahimi and Benjamin Recht, "Random Features for Large-Scale Kernel Machines", *in*: *Advances in Neural Information Processing Systems*, vol. 20, 2007.

[Sal+16] Ahmed H. Salamah, Mohamed Tamazin, Maha A. Sharkas, and Mohamed Khedr, "An enhanced WiFi indoor localization system based on machine learning", *in*: *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8.

[Sar+23] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G. Baraniuk, "WIRE: Wavelet Implicit Neural Representations", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 18507–18516.

[Sar+24] Hemanth Saratchandran, Sameera Ramasinghe, Violetta Shevchenko, Alexander Long, and Simon Lucey, "A sampling theory perspective on activations for implicit neural representations", *in*: *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, Vienna, Austria: JMLR.org, 2024.

[SB18] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, Second, The MIT Press, 2018.

[Sch86] R. Schmidt, "Multiple emitter location and signal parameter estimation", *in*: *IEEE Transactions on Antennas and Propagation* 34.*3* (Mar. 1986), pp. 276–280.

[Sch92] Jürgen Schmidhuber, "Learning to control fast-weight memories: An alternative to dynamic recurrent networks", *in*: *Neural Computation* 4.*1* (1992), pp. 131–139.

[SchHie21] Johannes Schmidt-Hieber, "The Kolmogorov-Arnold representation theorem revisited", *in*: *Neural Networks* 137 (2021), pp. 119–126.

[SCL24] Hemanth Saratchandran, Shin-Fang Chng, and Simon Lucey, "Analyzing the Neural Tangent Kernel of Periodically Activated Coordinate Networks", *in*: *arXiv preprint* (Feb. 2024).

[Sco15] David W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, Wiley, Mar. 2015.

[SDW17] Neev Samuel, Tzvi Diskin, and Ami Wiesel, "Deep MIMO detection", *in*: *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2017, pp. 1–5.

[SDW19] Neev Samuel, Tzvi Diskin, and Ami Wiesel, "Learning to Detect", *in*: *IEEE Transactions on Signal Processing* 67.*10* (2019), pp. 2554–2564.

[SE23] Nir Shlezinger and Yonina C. Eldar, "Model-Based Deep Learning", *in*: *Foundations and Trends® in Signal Processing* 17.*4* (2023), pp. 291–416.

[SEB22] Nir Shlezinger, Yonina C. Eldar, and Stephen P. Boyd, "Model-Based Deep Learning: On the Intersection of Deep Learning and Optimization", *in*: *IEEE Access* 10 (2022), pp. 115384–115398.

[SEB24] Phillip Stephan, Florian Euchner, and Stephan ten Brink, "Angle-Delay Profile-Based and Timestamp-Aided Dissimilarity Metrics for Channel Charting", *in*: *IEEE Transactions on Communications* 72.*9* (2024), pp. 5611–5625.

[Sha48] C. E. Shannon, "A mathematical theory of communication", *in*: *The Bell System Technical Journal* 27.*3* (1948), pp. 379–423.

[Shl+21] Nir Shlezinger, Jay Whang, Yonina C. Eldar, and Alexandros G. Dimakis, "Model-Based Deep Learning: Key Approaches and Design Guidelines", *in*: *2021 IEEE Data Science and Learning Workshop (DSLW)*, 2021, pp. 1–6.

[Shl+23] Nir Shlezinger, Jay Whang, Yonina C. Eldar, and Alexandros G. Dimakis, "Model-Based Deep Learning", *in*: *Proceedings of the IEEE* 111.*5* (2023), pp. 465–499.

[Shm+23] Dor H. Shmuel, Julian P. Merkofer, Guy Revach, Ruud J. G. van Sloun, and Nir Shlezinger, "Deep Root Music Algorithm for Data-Driven Doa Estimation", *in*: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[Shm+25] Dor Haim Shmuel, Julian P. Merkofer, Guy Revach, Ruud J. G. van Sloun, and Nir Shlezinger, "SubspaceNet: Deep Learning-Aided Subspace Methods for DoA Estimation", *in*: *IEEE Transactions on Vehicular Technology* 74.*3* (2025), pp. 4962–4976.

[Sil+16] David Silver et al., "Mastering the game of Go with deep neural networks and tree search", *in*: *Nature* 529.*7587* (Jan. 2016), pp. 484–489.

[Sil+18] David Silver et al., "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play", *in*: *Science* 362.*6419* (2018), pp. 1140–1144.

[Sit+20] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein, "Implicit neural representations with periodic activation functions", *in*: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7462–7473.

[SJ17] Krishnasamy T. Selvan and Ramakrishna Janaswamy, "Fraunhofer and Fresnel Distances: Unified derivation for aperture antennas", *in*: *IEEE Antennas and Propagation Magazine* 59.*4* (2017), pp. 12–15.

[SL17] Vincent Savaux and Yves Louët, "LMMSE channel estimation in OFDM context: a review", *in*: *IET Signal Processing* 11.*2* (2017), pp. 123–134.

[SM05] Petre G. Stoica and Randolph Moses, *Spectral analysis of signals*, ed. by Petre Stoica and Randolph L. Moses, Includes bibliographical references (p. 423-434) and index, Upper Saddle River, NJ: Pearson, Prentice Hall, 2005, 452 pp.

[Sol+19] Mehran Soltani, Vahid Pourahmadi, Ali Mirzaei, and Hamid Sheikhzadeh, "Deep Learning-Based Channel Estimation", *in*: *IEEE Communications Letters* 23.*4* (2019), pp. 652–655.

[Spe+99] M. Speth, S. A. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for wireless broad-band systems using OFDM. I", *in*: *IEEE Transactions on Communications* 47.*11* (1999), pp. 1668–1677.

[SS04] P. Stoica and Y. Selen, "Model-order selection: a review of information criterion rules", *in*: *IEEE Signal Processing Magazine* 21.*4* (2004), pp. 36–47.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.

[Stu+18] Christoph Studer, SaïD Medjkouh, Emre Gonultaş, Tom Goldstein, and Olav Tirkkonen, "Channel Charting: Locating Users Within the Radio Environment Using Channel State Information", *in*: *IEEE Access* 6 (2018), pp. 47682–47698.

[Sun+22] Qiang Sun, Huan Zhao, Jue Wang, and Wei Chen, "Deep Learning-Based Joint CSI Feedback and Hybrid Precoding in FDD mmWave Massive MIMO Systems", *in*: *Entropy* 24.*4* (2022).

[SZW19] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein, "Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations", *in*: *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[Tan+20] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng, "Fourier features let networks learn high frequency functions in low dimensional domains", *in*: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7537–7547.

[Tan+23] Sueda Taner, Maxime Guillaud, Olav Tirkkonen, and Christoph Studer, "Channel Charting for Streaming CSI Data", *in*: *2023 57th Asilomar Conference on Signals, Systems, and Computers*, 2023, pp. 1648–1653.

[TdSL00] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction", *in*: *Science* 290.*5500* (2000), pp. 2319–2323.

[Tel99] Emre Telatar, "Capacity of Multi-antenna Gaussian Channels", *in*: *European Transactions on Telecommunications* 10.*6* (1999), pp. 585–595.

[TI24] Deeb Assad Tubail and Salama Said Ikki, "Bayesian Cramer-Rao Bound, Extended and Unscented Kalman Filters Based Tracking Through Non-Ideal Transceivers in 5G and Beyond", *in*: *IEEE Transactions on Vehicular Technology* (2024), pp. 1–13.

[TPS25] Sueda Taner, Victoria Palhares, and Christoph Studer, "Channel Charting in Real-World Coordinates With Distributed MIMO", *in*: *IEEE Transactions on Wireless Communications* 24.*9* (2025), pp. 7286–7300.

[Tre+21] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt, "Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video", *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12939–12950.

[TV05] David Tse and Pramod Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.

[TW10] Joel A. Tropp and Stephen J. Wright, "Computational Methods for Sparse Solution of Linear Inverse Problems", *in*: *Proceedings of the IEEE* 98.*6* (2010), pp. 948–958.

[Vap00] Vladimir N. Vapnik, *The Nature of Statistical Learning Theory*, Springer New York, 2000.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is All you Need", *in*: *Advances in Neural Information Processing Systems*, ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, vol. 30, Curran Associates, Inc., 2017.

[Vas09] Volodymyr Vasylyshyn, "Direction of arrival estimation using ESPRIT with sparse arrays", *in*: *2009 European Radar Conference (EuRAD)*, 2009, pp. 246–249.

[Vil+24] Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn, "Position: will we run out of data? limits of LLM scaling based on human-generated data", *in*: *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, Vienna, Austria: JMLR.org, 2024.

[VK01] Jarkko Venna and Samuel Kaski, "Neighborhood Preservation in Nonlinear Projection Methods: An Experimental Study", *in*: *Artificial Neural Networks — ICANN 2001*, ed. by Georg Dorffner, Horst Bischof, and Kurt Hornik, Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 485–491.

[VVB88] B.D. Van Veen and K.M. Buckley, "Beamforming: a versatile approach to spatial filtering", *in*: *IEEE ASSP Magazine* 5.*2* (1988), pp. 4–24.

[Wan+17] Tianqi Wang, Chao-Kai Wen, Hanqing Wang, Feifei Gao, Tao Jiang, and Shi Jin, "Deep learning for wireless physical layer: Opportunities and challenges", *in*: *China Communications* 14.*11* (2017), pp. 92–111.

[Wan+19] Tianqi Wang, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li, "Deep Learning-Based CSI Feedback Approach for Time-Varying Massive MIMO Channels", *in*: *IEEE Wireless Communications Letters* 8.*2* (2019), pp. 416–419.

[WHD21] Xiuhong Wei, Chen Hu, and Linglong Dai, "Deep Learning for Beamspace Channel Estimation in Millimeter-Wave Massive MIMO Systems", *in*: *IEEE Transactions on Communications* 69.*1* (2021), pp. 182–193.

[Wik25a] Wikipedia, *Fresnel zone — Wikipedia, The Free Encyclopedia*, 2025.

[Wik25b] Wikipedia, *Guglielmo Marconi — Wikipedia, The Free Encyclopedia*, 2025.

[Wik25c] Wikipedia, *List of wireless network technologies — Wikipedia, The Free Encyclopedia*, 2025.

[WLH19] Liuli Wu, Zhang-Meng Liu, and Zhi-Tao Huang, "Deep Convolution Network for Direction of Arrival Estimation With Sparse Prior", *in*: *IEEE Signal Processing Letters* 26.*11* (Nov. 2019), pp. 1688–1692.

[WM97] D.H. Wolpert and W.G. Macready, "No free lunch theorems for optimization", *in*: *IEEE Transactions on Evolutionary Computation* 1.*1* (1997), pp. 67–82.

[WSJ18] Chao-Kai Wen, Wan-Ting Shih, and Shi Jin, "Deep Learning for Massive MIMO CSI Feedback", *in*: *IEEE Wireless Communications Letters* 7.*5* (2018), pp. 748–751.

[Wym+20] Henk Wymeersch, Jiguang He, Benoit Denis, Antonio Clemente, and Markku Juntti, "Radio Localization and Mapping With Reconfigurable Intelligent Surfaces: Challenges, Opportunities, and Research Directions", *in*: *IEEE Vehicular Technology Magazine* 15.*4* (2020), pp. 52–61.

[Xia+21] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim, "Space-time Neural Irradiance Fields for Free-Viewpoint Video", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9421–9431.

[Xue+23] Qiulin Xue, Chao Dong, Xiangjun Li, Jianzhong Yi, and Kai Niu, "Integrated Deep Implicit CSI Feedback and Beamforming Design for FDD mmWave Massive MIMO Systems", *in*: *IEEE Wireless Communications Letters* 12.*1* (2023), pp. 119–123.

[Yan+21] Jie Yang, Shi Jin, Chao-Kai Wen, Jiajia Guo, Michail Matthaiou, and Bo Gao, "Model-Based Learning Network for 3-D Localization in mmWave Communications", *in*: *IEEE Transactions on Wireless Communications* 20.*8* (2021), pp. 5449–5466.

[Yar+20] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance", *in*: *Advances in Neural Information Processing Systems* 33 (2020), pp. 2492–2502.

[Yas+22] Taha Yassine, Luc Le Magoarou, Stéphane Paquelet, and Matthieu Crussière, "Leveraging triplet loss and nonlinear dimensionality reduction for on-the-fly channel charting", *in*: *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communications*, 2022, pp. 1–5.

[Yas+23] Taha Yassine, Baptiste Chatelier, Vincent Corlay, Matthieu Crussière, Stéphane Paquelet, Olav Tirkkonen, and Luc Le Magoarou, "Model-Based Deep Learning for Beam Prediction Based on a Channel Chart", *in*: *2023 57th Asilomar Conference on Signals, Systems, and Computers*, 2023, pp. 1636–1640.

[Yas+24] Taha Yassine, Luc Le Magoarou, Matthieu Crussière, and Stéphane Paquelet, "Optimizing Multicarrier Multiantenna Systems for LoS Channel Charting", *in*: *IEEE Transactions on Wireless Communications* 23.*10* (2024), pp. 14702–14714.

[Yim08] Jaegeol Yim, "Introducing a decision tree-based indoor positioning technique", *in*: *Expert Systems with Applications* 34.*2* (2008), pp. 1296–1302.

[YLM22] Taha Yassine and Luc Le Magoarou, "mpNet: Variable Depth Unfolded Neural Network for Massive MIMO Channel Estimation", *in*: *IEEE Transactions on Wireless Communications* 21.*7* (2022), pp. 5703–5714.

[Yuc+22] Gizem Yuce, Guillermo Ortiz-Jimenez, Beril Besbinar, and Pascal Frossard, "A Structured Dictionary Perspective on Implicit Neural Representations", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[YuHsu25] Joey Yu Hsu, *Kernel Density Estimation*, https://bookdown.org/redwing1304/math/matplotlib.html, 2025.

[Zha+23] Xiaopeng Zhao, Zhenlin An, Qingrui Pan, and Lei Yang, "NeRF2: Neural Radio-Frequency Radiance Fields", *in*: *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023.

[ZK13] P. Zhu and A.V. Knyazev, "Angles between subspaces and their tangents", *in*: *Journal of Numerical Mathematics* 21.*4* (Jan. 2013).

[ZRL21] Jianqiao Zheng, Sameera Ramasinghe, and Simon Lucey, "Rethinking Positional Encoding", *in*: *ArXiv* (July 2021).

COLLEGE DOCTORAL BRETAGNE | MATHS, TELECOMS INFORMATIQUE, SIGNAL SYSTEMES, ELECTRONIQUE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Titre : Apprentissage basé sur les modèles pour les systèmes sans fil de grande dimension

Mot clés : Apprentissage automatique basé sur un modèle, Représentations neuronales implicites, Radiolocalisation, Imperfections matérielles, Compression de canal

Résumé : Le domaine des communications sans fil s'est traditionnellement appuyé sur des techniques de traitement du signal, basées sur des modèles mathématiques, pour décrire et optimiser la transmission sans fil de l'information. Lorsque les hypothèses sous-jacentes à ces modèles ne sont pas pleinement satisfaites, les méthodes conventionnelles présentent souvent des performances sous-optimales. Récemment, des approches basées sur l'apprentissage automatique ont été utilisées pour traiter ce problème en raison de leurs fortes capacités d'adaptation. Néanmoins, ces méthodes introduisent également des problématiques importantes, notamment en termes d'interprétabilité et de complexité.

Cette thèse se concentre sur l'utilisation du paradigme de l'apprentissage automatique basé sur les modèles, dans les systèmes sans fil de communication ou de détection. Ce paradigme s'appuie sur les modèles mathématiques utilisés dans les approches conventionnelles de traitement du signal pour structurer, initialiser ou optimiser des méthodes d'apprentissage, combinant ainsi la structure et l'interprétabilité des méthodes conventionnelles, à l'adaptabilité de l'apprentissage automatique. Ce paradigme est étudié à travers trois cas d'usage spécifiques : l'apprentissage de la fonction position-canal, la compensation des imperfections matérielles et la compression de canal.

Title: Model-based learning for high-dimensional wireless systems

Keywords: Model-based machine learning, Implicit Neural Representations, Radio Localization, Hardware Impairments, Channel compression

Abstract: The field of wireless communications has traditionally relied on signal processing techniques, based on mathematical models, to describe and optimize wireless transmission of information. When the assumptions underlying these models are not fully met, conventional methods often yield suboptimal performance. Recently, machine learning methods have been used to address this issue, owing to their strong adaptation capabilities. However, these methods also introduce significant challenges in terms of interpretability and complexity.

This thesis focuses on the use of the model-based machine learning paradigm in wireless communication or sensing systems. This paradigm leverages mathematical models used in classical signal processing approaches to structure, initialize, or optimize learning methods, thereby combining the structure and interpretability of classical methods with the adaptability of machine learning. This paradigm is studied through three specific use cases: location-to-channel mapping learning, hardware impairments compensation, and channel compression.