THE UNIVERSITY *of* EDINBURGH

Channel Estimation and Signal Detection with Model-Driven Deep Learning for Massive Multiuser MIMO-OFDM Systems

Changjiang Liu

A thesis submitted for the degree of
DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF EDINBURGH

January 2023

Abstract

The emerging fifth generation (5G) and beyond wireless systems raise requirements on improved coverage, lower latency, higher data rates, and energy efficiency. These improvements can be provided by massive Multiple-Input Multiple-Output (MIMO), which is one of the essential technologies of 5G. However, due to a large number of antennas and radio frequency chains, the receiver design for massive MIMO systems becomes more challenging. Moreover, non-linear effects in communication systems like hardware impairment are hard to be modeled. Since conventional solutions struggle to address these challenges, deep learning (DL) is considered as a promising approach for the sixth generation cellular systems. The application of DL in the physical layer is still in a nascent stage. Most prior DL-based receivers are purely data-driven and aim mainly for performance improvement. By contrast, model-driven DL-based massive MIMO receivers can potentially achieve lower complexity, better interpretability and robustness by introducing expert knowledge, but this is not yet well-investigated in the literature. This PhD thesis focuses on designing low-complexity channel estimation and signal detection schemes with the application of model-driven DL techniques for massive MIMO orthogonal frequency-division multiplexing (OFDM) systems.

Firstly, a novel model-aided DL-based OFDM receiver, which integrates a convolutional neural network (CNN)-based channel estimator and a signal detection neural network (SD-NN), is proposed. By exploiting channel correlations in the time and frequency domain, the proposed CNN-based scheme can efficiently denoise and refine the image-like low-resolution channel with time-variant and frequency-selective

effects. Then, the explicit channel information from the CNN is processed by a model-based equalizer, and the output is used as a proper initialization for SD-NN. Moreover, several data preprocessing and model tuning strategies are developed to improve detection performance. As a result, unlike the data-driven solutions viewing the whole receiver as a black box, the proposed model-aided DL architecture achieves lower complexity and faster convergence with only a small number of pilots.

Secondly, considering the complexity problem of generic NN architectures in massive MIMO-OFDM systems, an efficient deep unfolding (DU)-based detection network is developed. Based on the domain knowledge of MIMO detection, the alternating direction method of multipliers (ADMM)-based network skeleton is first derived. To get a specialized architecture with improved model flexibility for DU-based MIMO-OFDM detection, the over-relaxation parameter and an additional step size are added to the network skeleton. With the help of DU techniques, the sets of trainable parameters can be optimized explicitly for different subcarriers to enhance the detection performance under realistic channels with severe spatial and frequency correlations. Furthermore, a differentiable projection function is designed to enable learning-based parameter optimization. Compared to existing baselines in the literature, the proposed approach can provide a better performance-complexity trade-off, especially for the cases of high user load and real-world correlated channels.

Thirdly, for high-loaded massive MIMO systems with large numbers of users, a frequency-orthogonal pilot scheme is designed to save time resources used for pilot transmission. By a special subtractive residual layer, the proposed denoising CNN learns the denoising mapping to eliminate channel noise in the delay domain instead of learning the labeled channel matrices directly. To further improve estimation performance, a residual CNN is proposed to exploit spatial-frequency correlations of channels and refine the output of the denoising CNN. With the help of a customized

fast Fourier transform layer, these two CNNs can be jointly trained across different domains, resulting in an end-to-end channel estimation network. This dual CNN-based estimator is shown to achieve state-of-the-art performance and fast convergence with lower complexity than baseline approaches.

Finally, the research is extended to massive MIMO-OFDM systems with low-precision analog-to-digital converters (ADCs). To accurately detect the received signals with severe non-linear distortions under such complex systems, a novel model-driven detection network is proposed. Since well-established architectures like ADMM can not handle errors of coarse quantization, a flexible non-linear estimator is first derived to replace the general $\boldsymbol{x}$-update of ADMM. Specifically, the scalar step size is upgraded to a learnable vector used as the multiplicative gradient correction, and an additive gradient correction is also added. Correspondingly, a specialized network skeleton with multiple trainable parameters and an adaptive proximal operator is designed. By fusing the model-based architecture and data-driven techniques, the proposed scheme shows superior detection performance and robustness in coarsely quantized massive multiuser MIMO-OFDM systems.

Lay Summary

Within 40 years, the cellular communication system has experienced extensive development from the first to the fifth generation (5G). Unlike the past generations, the current 5G focus on not only human-to-human but also human-to-thing and thing-to-thing communications. Advanced wireless applications, such as Internet of Things and autonomous driving, have posed unprecedented demands on the number of connections, data rate and latency. To meet these performance criteria, massive multiple-input multiple-output (MIMO) is proposed as one of the critical technologies of 5G wireless networks. However, 5G wireless systems with massive MIMO are much more complicated than conventional systems. Moreover, the new requirements to dynamically adapt to complex mobile environments and handle a huge amount of wireless data challenge traditional schemes. Based on its great success, deep learning (DL) provides a powerful alternative that can address the challenges in future communication systems, which is also considered by the sixth generation standards.

Over the past decade, DL has been widely applied to some tasks in upper layers of wireless communication systems, while its application in receiver design is still in a nascent stage. In the existing literature, most receivers are based on purely data-driven DL architectures, which have high complexity and also require considerable computing resources, memory, training time and data. Thus, this thesis aims to design model-driven DL-based receivers for the next generation massive MIMO multi-carrier systems. By combining DL technologies with expert knowledge, model-driven schemes are able to achieve a relatively ideal trade-off between performance and complexity. For channel estimation, conventional algorithms are used to aid generic

DL networks for better convergence. Moreover, specialized model-based architectures are proposed for efficient massive MIMO detection. In this thesis, the robustness of the proposed solutions to realistic channels, high user load and low-precision quantization, are also highlighted.

Declaration of Originality

I hereby declare that this thesis was composed and originated by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Changjiang Liu

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Tughrul Arslan, for his guidance and support during the whole period of my PhD. His encouragement and patience help me to become an independent researcher and perform world-class research in an emerging field. I also thank him for his vision and advice on my career. I am short of words to thank my second supervisor Prof. John Thompson, who has been very supportive and professional throughout my PhD study. During our monthly catchup meetings, he has always provided constructive suggestions and feedback on my research project and academic writing. I will always be grateful to my supervisors for providing unending support and motivation in my research work and personal development. I have learned a lot from them.

A very special thanks to my beloved family members, especially my dearest parents, Junliang Liu and Xiaoli Kang. I would like to dedicate this thesis to them for their endless understanding, encouragement, and love throughout my life. Without them, I might not be able to finish this long journey. I would also be grateful to my late grandparents for their genuine care.

I am thankful to all my colleagues in the EWireless Research Group and the Scottish Microelectronics Centre at the University of Edinburgh. They create a positive and friendly research atmosphere. I truly appreciate Yichen Du, Rahmat Ullah, and Stefan Brennsteiner, who often had inspiring discussions with me about the research work and daily life. I would also like to thank all my dear friends in Edinburgh. Their care and emotional support always motivate me and make my life cheerful. I will remember the time we spent together before and after the COVID-19 pandemic.

Last but not least, a special thanks goes to my beloved girlfriend, Yue Wang, for her constant love, support, and sacrifices. She has always stood by my side no matter where we are, removed my stress, and encouraged me to keep moving. I can not thank you enough for accompanying me in both Zhengzhou and Edinburgh.

Contents

List of Figures

List of Tables

List of Symbols

$\boldsymbol{b}$	Binary symbol
b_k	Trainable offset in the proposed prox()
c	Bias in neural network layer
d	Distance between the transmitter and the receiver
$\mathcal{F}$	FFT Matrix
$g(\cdot)$	Indicator function
$\boldsymbol{G}$	Equalization matrix
G_{tx}, G_{rx}	Transmit antenna gain, receive antenna gain
h	Channel response
$\boldsymbol{H}$	Channel matrix
$\widehat{\boldsymbol{H}}$	Estimate of the channel
k	Iteration or layer index
K	Kernel size
L	Number of channel paths
$\mathcal{L}(\cdot)$	Augmented Lagrangian
M	Number of single-antenna users
$\mathcal{M}$	Finite quantization alphabet
$\boldsymbol{n}$	Gaussian noise vector
N	Number of BS antennas
P_{tx}, P_{rx}	Transmit power, receive power
PL	Path loss
q_i	Quantization label

$\mathcal{Q}(\cdot)$	Lloyd-Max quantizer
S	Number of training samples
T	Number of iterations or layers
$\boldsymbol{u}$	Scaled dual variable
W	Number of subcarriers
$\boldsymbol{x}$	Input vector or transmit symbol vector
$\boldsymbol{y}$	Output vector or receive symbol vector
$\boldsymbol{z}$	Iterative variable of ADMM
π	Over-relaxed parameter
α	Over-relaxed parameter
β	Trainable offset in the soft-sign function
δ	Extra trainable step size in the proposed MMO-Net
ε, θ	Trainable scalar parameters in the proposed QMMO-Net
λ	Wavelength
λ_k	Learnable smoothing coefficient of k-th iteration in the proposed proj()
ρ	The only step size in generic ADMM iterations
$\boldsymbol{\rho}_k$	Trainable vector parameter in the k-th layer of QMMO-Net
σ	Standard deviation
χ	Constellation set
ω	Subcarrier index

Notations

a or A	Scalar
$\|a\|$	Absolute value of a
$\boldsymbol{a}$	Vector
$\boldsymbol{A}$	Matrix
$\boldsymbol{A}^{\boldsymbol{T}}$	Transpose of A
$\boldsymbol{A}^{\boldsymbol{H}}$	Conjugate transpose of A
$\boldsymbol{A}^{-1}$	Inverse of A
$\boldsymbol{A}^{\dagger}$	Pseudo inverse of A
$[\boldsymbol{A}]_{i,:}$	i-th row of A
$[\boldsymbol{A}]_{:,k}$	k-th colunm of A
$[\boldsymbol{A}]_{i,k}$ or $[\boldsymbol{A}]^{i,k}$	The entry at the i-th row and k-th column in A
$\|\|a\|\|_p$	p-norm of a
$\mathrm{diag}\,(\boldsymbol{A})$	Matrix generated by the diagonal elements of A
$\mathcal{CN}\,(a,b)$	Complex Gaussian vector with mean a and covariance b
$\mathbb{R}$	Set of real numbers
$\mathbb{C}$	Set of complex numbers
$\mathbb{R}^{A\times B}$	Set of $A\times B$ matrices with real entries
$\mathbb{C}^{A\times B}$	Set of $A\times B$ matrices with complex entries
$\boldsymbol{X}\in\mathbb{C}^{A\times B}$	$A\times B$ matrix $\boldsymbol{X}$ with complex entries
$\boldsymbol{I}_N$	$N\times N$ identity matrix
$\Re\{\cdot\}$	Real part of a complex variable
$\Im\{\cdot\}$	Imaginary part of a complex variable

$a \odot b$	Hadamard product of a and b
$\frac{\partial f}{\partial x}$	Partial derivative of a function f with respect to the variable x
$\sum_{i=1}^{N} x_i$	Sum the values from x_1 to x_N
$\nabla f(x)$	Gradient of a function f at point x
$\log(\cdot)$	Logarithmic function
$\text{proj}(\cdot)$	Projection function
$\text{prox}(\cdot)$	Proximal operator
$\max(\cdot)$	Maximum value function
$\min(\cdot)$	Minimum value function
$\mathcal{R}(\cdot)$	Residual mapping
$\mathcal{H}(\cdot)$	Denoising mapping

Abbreviations and Acronyms

1D	One dimensional
2D	Two dimensional
3D	Three dimensional
3G	Third generation
3GPP	3rd Generation Partnership Project
5G	Fifth generation
6G	Sixth generation
ADC	Analog-to-digital converter
ADMM	Alternating Direction Method of Multipliers
AE	Autoencoder
AI	Artificial Intelligence
AMI	Approximate matrix inversion
AMP	Approximate message passing
AWGN	Additive white gaussian noise
BER	Bit error rate
BPSK	Binary Phase Shift Keying
BS	Base station
CCI	Co-channel interference
CE-NN	Channel estimation neural network
CFR	Channel frequency response
CG	Conjugate gradient
CIR	Channel impulse response

CNN	Convolutional Neural Network
CP	Cyclic prefix
CSI	Channel state information
DFT	discrete Fourier transform
DL	Deep Learning
DNN	Deep Neural Network
DU	Deep Unfolding
FBS	Forward-backward splitting
FC	Fully connected
FFT	Fast Fourier Transform
FLOPs	Floating-point operations
IFFT	Inverse Fast Fourier Transform
i.i.d	Independent and identically distributed
ISI	Inter-symbol interference
LR	Learning rate
LS	Least squares
LSTM	Long short-term memory
LTE	Long-Term Evolution
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MLP	Multilayer Perceptron
MMO-Net	Neural network for massive MIMO-OFDM detection
MMSE	Minimum mean square error
MSE	Mean square error
MU	Multiuser
NLOS	Non-line-of-sight
NN	Neural network

OAMP	Orthogonal approximate message passing
OFDM	Orthogonal Frequency-division Multiplexing
PHY	Physical layer
QAM	Quadrature Amplitude Modulation
ReLU	Rectified Linear Unit
RF	Radio frequency
Rx	Receiver
Tx	Transmitter
SGD	Stochastic gradient descent
SIMO	Single-Input Multiple-Output
SISO	Single-Input Single-Output
SNR	Signal-to-noise ratio
S/P	Serial-to-parallel
tanh	Hyperbolic tangent
ZF	Zero forcing

Chapter 1

Introduction

This PhD thesis aims to exploit deep learning (DL)-based technologies to enable some key improvements of future wireless communication systems like the fifth generation (5G) and beyond. Even if deep learning has garnered growing interest in physical layer communications due to its great successes in image and speech processing, how the DL-based solutions can be properly applied to 5G-related massive multiple-input multiple-output orthogonal frequency-division multiplexing (MIMO-OFDM) systems is still an open issue. Our main focus is to develop advanced receiver schemes including channel estimation and signal detection by fusing the power of data-driven DL techniques and communication domain knowledge. In this chapter, Section 1.1 first explains the research motivation. The objectives and key contributions of the research work are summarized in Section 1.2 and Section 1.3, respectively. Section 1.4 presents the outline of this thesis.

1.1 Research Motivation

Enabled by Long-Term Evolution (LTE), various applications like mobile games and short-form videos have triggered a dramatically increasing number of mobile devices and data traffic. According to Cisco's networking index forecast, there will be around 4.8 billion Internet users and 28.5 billion devices and connections by 2022 [1]. The explosion of wireless devices and emerging wireless applications, such as diverse

intelligent terminal access, virtual reality, and Internet of things (IoT), has propelled the development of wireless communication into 5G [2]. The emerging 5G wireless communication system is expected to provide ultra-low latency, higher throughput, reliability, and spectral efficiency.

As one of the key technologies of 5G, massive MIMO is capable of providing significant improvements to energy efficiency, spectral efficiency, and robustness of the system [3]. As another widely-used technology in modern digital communications, OFDM is a multi-carrier modulation scheme that can efficiently address frequency-selective fading in wireless channels. Thus, the combination of MIMO and OFDM has drawn attention to many researchers. However, the design of massive MIMO-OFDM systems is complicated. The resurgence of deep learning techniques offers an opportunity to revisit the massive MIMO-OFDM receiver design problem [4]. DL has been widely applied to the upper layers of wireless communication systems. Within several years, DL has started to regain attention in signal processing applications in the physical layer [5]. However, there are still many challenges ahead to realize the full potential of DL-based physical layer technologies like accurate channel estimation and low-complexity detection of massive MIMO. These challenges can be summarized as follows:

1. Channel modeling and estimation in complex scenarios. Conventional MIMO channel models fail to capture new features introduced by the increase of antennas in massive MIMO systems [6]. Moreover, without accurate prior channel statistics, the performance of conventional channel estimation schemes like MMSE will degrade especially in low signal-to-noise ratios (SNRs).

2. Complexity and robustness of signal detection. As the number of antennas grows, the complexity of the matrix inversion and conventional data detection schemes increases significantly. When the number of transmit antennas is comparable to the number of BS antennas, the design of detectors with reasonable complexity becomes more challenging [7]. Furthermore, non-linear imperfections in massive MIMO-OFDM systems like low-precision quantization are difficult to be handled. Thus, it is desirable to develop a robust receiver scheme that can achieve a good trade-off between performance and complexity.

3. Training overhead and interpretability of DL-based schemes. Most of the existing DL-based receivers are purely data-driven, which have large numbers of trainable parameters, especially for massive MIMO-OFDM systems. The performance of well-trained data-driven solutions could be competitive, but they require sufficient computing resources and a huge amount of labeled data, both of which are rarely found in wireless communication devices [8]. In addition, data-driven schemes use the generic neural network (NN) as a black box, which makes their architectures unexplainable. Due to these limitations, it is difficult to implement data-driven DL solutions in some practical applications.

1.2 Objectives and Contributions

1.2.1 Research Objectives

The main aim of this thesis is to develop efficient receivers that can improve the accuracy of the channel estimation and decrease the complexity of signal detection for massive MIMO-OFDM systems by utilizing state-of-the-art deep learning technologies. The fast-increasing demand for wireless connectivity requires a lot of innovations in the sixth generation (6G) communication networks, and DL tools will play a key

role in solving problems in the wireless domain [8]. However, most prior data-driven DL-based receivers mainly focus on performance improvement rather than complexity reduction. These schemes require a long training time in addition to a huge volume of training data. These problems motivate us to combine DL networks with expert knowledge, i.e. model-driven neural networks. By designing the NN architectures aided by conventional algorithms or integrating DL techniques into existing signal processing schemes, the system performance and robustness to non-linear distortions will be improved. Meanwhile, with the help of communication domain knowledge, model-driven DL-based receivers can be trained faster and achieve a reasonable trade-off between performance and complexity.

1.2.2 Key Contributions

The key contributions of this thesis are summarized as follows:

1. A novel OFDM receiver, which includes a channel estimation neural network (CE-NN) and a signal detection neural network (SD-NN), is proposed. By exploiting the learning and generalization capacity of convolutional neural networks (CNN), CE-NN can estimate the two-dimensional response of time-variant and frequency-selective channels using only a small number of pilots. With the help of the CE-NN and specialized training strategies, the SD-NN in RecNet outperforms the end-to-end DL-based receivers like [9]. Compared with the purely data-driven solutions in the literature, the proposed CE-NN and SD-NN achieve faster convergence and require fewer training data with the aid of model-based algorithms. Moreover, the robustness of SD-NN is evaluated for different SNRs and lengths of the cyclic prefix.

2. An efficient model-driven detection network is developed for massive multiuser (MU)-MIMO-OFDM systems, named MMO-Net. First, based on a general alternating direction method of multipliers (ADMM) architecture, we derive a MIMO detector used as the network skeleton. Then, the over-relaxation parameter and an extra step size are added to the detector to increase the model flexibility. To better adapt to realistic correlated channels, we unfold the algorithm iterations in a layer-wise neural network and jointly learn its parameters explicitly for different subcarriers from the training data. In addition, a specific differentiable projection is designed to enable gradient descent-based optimization. To demonstrate the superior performance and robustness of MMO-Net, we implement several state-of-the-art detection networks into the massive MU-MIMO-OFDM system with high user load and real-world channels considered by few prior works.

3. Three CNN-based channel estimators, i.e., a denoising CNN (D-CNN), a residual spatial-frequency CNN (RSF-CNN), and a DRSF-CNN are proposed for MU-MIMO-OFDM systems. Instead of learning the labeled channel matrices directly, D-CNN learns the denoising mapping via a special subtractive residual layer to eliminate channel noise in the delay domain. To further enhance the performance of D-CNN, we propose RSF-CNN to exploit spatial and frequency correlations of channels. Furthermore, a fast Fourier transform (FFT) layer is developed to enable end-to-end training of D-CNN and RSF-CNN across different domains, resulting in DRSF-CNN. The number of trainable parameters, computational complexity and estimation performance for the existing CNN-based estimators and the proposed schemes are also compared.

4. To fuse the advantages of model-based algorithms and data-driven DNNs, we propose a model-driven network, QMMO-Net, by introducing deep unfolding (DU) techniques in quantized massive MIMO-OFDM detection tasks. Unlike the case of infinite-precision systems, well-established model-based architectures like ADMM can not handle severe distortions from low-precision quantization, even with the help of DU tools. Thus, we derive an architecture specialized for the detection of coarsely quantized signals. First, a flexible non-linear estimator with a trainable vector is developed as the new $\boldsymbol{x}$-update step, and an additive gradient correction is added. Correspondingly, the ADMM-based network skeleton and learnable proximal operator are redesigned to improve detection performance further.

1.3 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2

This chapter provides an essential background for the thesis and introduces relevant techniques that are helpful to understand the following technical chapters. Firstly, the fading phenomenon in the wireless channel is explained. Secondly, an overview of massive MIMO-OFDM systems is presented, which includes MIMO channel models, the benefits and challenges of massive MIMO-OFDM systems, and the techniques for MIMO-OFDM detection. Then, the fundamentals and categories of deep learning algorithms are illustrated in detail. Finally, the applications of DL in physical layer communications, especially in channel estimation and signal detection, are discussed.

Chapter 3

This chapter studies DNN-based techniques for the OFDM receiver including channel estimation and signal detection. First, the formulation of the OFDM system used in this chapter is presented. Next, the proposed channel estimator including the lattice-type pilot, model-based initialization, and the CE-NN, is explained. Then, the technical challenges, the architecture of the DL-based OFDM receiver consisting of CE-NN and SD-NN, and the corresponding strategies of data preprocessing and model tuning are described in detail. The work in this chapter is mainly based on [10] and [11].

Chapter 4

This chapter comprehensively discusses the development of deep unfolding (DU)-based detection networks. First, the model of a massive MU-MIMO-OFDM system for 5G and beyond is introduced. Next, a MIMO detector is derived based on the generic ADMM architecture. Then, the proposed MMO-Net consisting of a novel network skeleton, a set of trainable parameters, and a differentiable projection function is described. Finally, the comprehensive performance evaluation and comparison of several state-of-the-art DU-based detectors are provided in the case of high user load and realistic channels. This chapter is partly based on the work in [12].

Chapter 5

This chapter investigates the channel estimation problem in massive MU-MIMO-OFDM systems. First, the system model and the prior works on DL-based channel estimation are discussed. Next, an optimal pilot scheme for highly loaded MU-MIMO systems and a D-CNN with an additive residual layer for channel denoising are proposed. Then, the network architecture of the DRSF-CNN, which includes a D-CNN, a customized FFT layer and a residual RSF-CNN, is described in detail. Finally, the performance and complexity of the proposed schemes and baselines are compared. This chapter is based on our work in [13].

Chapter 6

This chapter introduces DU techniques into the detection task for coarsely quantized massive MIMO-OFDM systems. First, the system structure and the quantizer used in this chapter are presented. Next, the technical challenges and the proposed model-driven detector specialized for the massive MU-MIMO-OFDM system with low-precision quantization are described. Then, the numerical results are provided to demonstrate the performance of the proposed detection network. This chapter is in part based on [14].

Chapter 7

This chapter concludes the research works conducted in this thesis and summarises the main contributions. The limitations and potential future research directions are also discussed.

1.4 Publications Arising from the Research

Conference Papers:

1. **Changjiang Liu**, and Tughrul Arslan, "RecNet: Deep learning-based OFDM receiver with semi-blind channel estimation," in 2020 IEEE International Symposium on Circuits and Systems (ISCAS), 2020, pp. 1–4.
2. **Changjiang Liu**, John Thompson, and Tughrul Arslan, "A Deep Unfolding Network for Massive Multi-user MIMO-OFDM Detection," in 2022 IEEE Wireless Communications and Networking Conference (WCNC), 2022, pp. 2405-2410.
3. **Changjiang Liu**, John Thompson, and Tughrul Arslan, "Deep Unfolding-based Detection for Quantized Massive MU-MIMO-OFDM Systems," in 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), 2022, pp. 1-5.

Journal Papers:

1. **Changjiang Liu**, John Thompson, and Tughrul Arslan, "OFDM Receivers with Semi-Blind Channel Estimation based on Deep Neural Networks," IEEE Transactions on Cognitive Communications and Networking, 2022. (Under Review)
2. **Changjiang Liu**, John Thompson, and Tughrul Arslan, "Deep Learning Based Receivers for Massive MIMO OFDM Systems with High User Load," IEEE Transactions on Wireless Communications, 2022. (Under Review)

Chapter 2

Background

This chapter presents a basic technical background for this thesis. This chapter starts by introducing the basic knowledge of wireless channels, including path loss, shadowing, and small-scale fading. Next, an overview of massive multiple-input multiple-output (MIMO) orthogonal frequency-division multiplexing (OFDM) systems is provided, which includes the key role of MIMO-OFDM systems in the era of the fifth generation (5G), the advantages and technical challenges of using massive MIMO, and several relevant signal detection algorithms. Then, basic concepts of artificial intelligence (AI), machine learning (ML) and deep learning (DL), as well as their relation, are described. The forward propagation and backward propagation processes are explained to help understand the fundamentals of DL, followed by categories of DL architectures. Finally, DL applications in physical layer communications are discussed, and the DL-based schemes for channel estimation and signal detection are highlighted.

2.1 Overview of Wireless Channel

The performance of wireless communication systems is mainly dependent on the wireless channel environment. Thus, the study of wireless channels is the basis for the research of channel estimation and signal detection algorithms. In the process of propagation, a wireless signal is affected by many factors, such as the distance

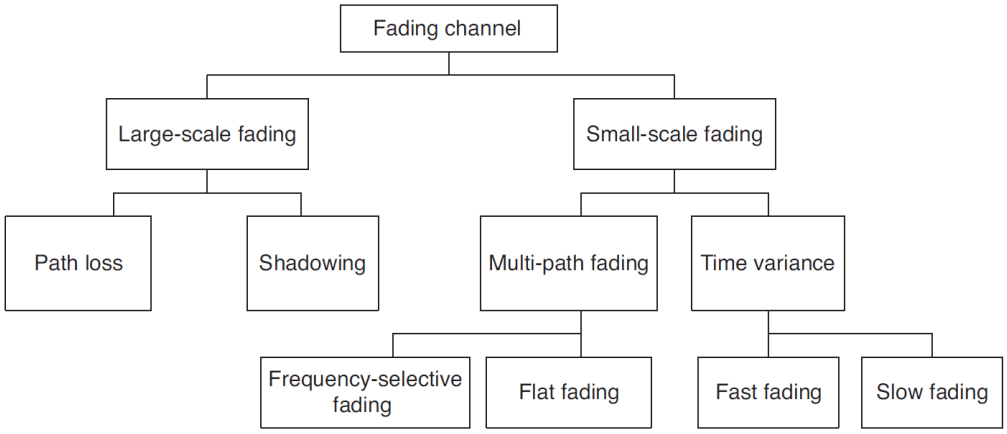


Figure 2.1: Classification of fading channels.

between the transmitter and receiver and the propagation environment [15]. In contrast with the additive noise, fading is characterized as a non-additive signal disturbance in the wireless channel. This section briefly introduces two different types of the fading phenomenon in a wireless channel, i.e., large-scale fading and small-scale fading.

2.1.1 Large-scale Fading

Large-scale fading occurs as mobile devices travel through a large distance. It can be characterized by path loss and shadowing. On the other side, when mobile devices travel short distances, the constructive and destructive interference of multipath signals leads to rapid fluctuations of signal power [16]. This phenomenon is called small-scale fading. The classification of fading channels is illustrated in Fig. 2.1.

2.1.1.1 Path Loss

For wireless communications, path loss refers to the energy attenuation of a signal caused by the propagation medium in the transmission process, which is a function of the distance d between the transmitter and receiver. When there are no obstructions between the transmitter and receiver in the line-of-sight environment, a simple free-space path loss $PL_{FS}(d)$ in decibels (dB) is given by:

$$PL_{FS}(d)[dB] = 10\log\left(\frac{P_{tx}}{P_{rx}}\right) = 10\log\left(\frac{(4\pi d)^2}{G_t G_r \lambda^2}\right) \tag{2.1}$$

where P_{tx} and P_{rx} are the transmit and receive power, separately. P_{rx} can be expressed by the well-known Friis Formula [17] as:

$$P_{rx}(d) = P_{tx}\frac{G_{tx}G_{rx}\lambda^2}{(4\pi d)^2} \tag{2.2}$$

where G_{tx} and G_{rx} are the transmit and receive antenna gains, respectively. λ represents the wavelength of radiation.

Similar to the free-space case mentioned above, the average power of received signals in other real-world environments also logarithmically decreases with the distance d. Thus, by adding the environment-dependent path loss exponent α [15] to the free-space path loss (2.1), a more generalized model for path loss can be described as follows:

$$PL(d)[dB] = PL_{FS}(d_0) + 10\alpha\log\left(\frac{d}{d_0}\right) \tag{2.3}$$

where d_0 is the reference distance which should be properly set for different scenarios.

2.1.1.2 Shadowing

In addition to path loss, large-scale fading also includes shadowing caused by large random objects along the path of the signal, such as buildings and vegetation. Let χ_{sh} (in dB) represent the shadowing effect, which is a Gaussian random variable with a standard deviation σ_{sh}. With consideration of shadowing effects, equation (2.3) becomes:

$$PL(d)[dB] = PL_{FS}(d_0) + 10\alpha\log\left(\frac{d}{d_0}\right) + \chi_{sh} \tag{2.4}$$

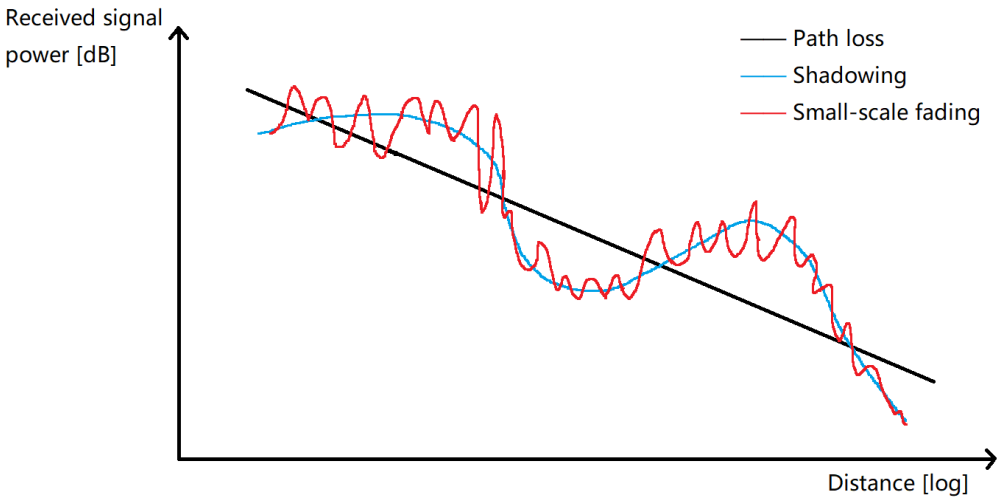


Figure 2.2: The received signal power affected by path loss, shadowing, and small-scale fading versus distance.

As shown in this formula, the signal power received at the places with the same distance d from the transmitter may be different due to shadowing. In practice, the shadowing model (2.4) is helpful for a more realistic situation.

2.1.2 Small-scale Fading

Small-scale fading is the rapid fluctuations of signal power caused by the short-distance movement of mobile users, which is due to the multipath propagation of wireless signals. The transmitted signal usually travels to the receiver via multiple paths since it is scattered or reflected by the objects located around its path. Due to their different paths, these scattered signals have different amplitudes and phases at the receiver. The differences of multipath signals in phases may result in the variation of the total received power [18]. Fig. 2.2 depicts three components of the channel response.

One of the most typical and important small-scale fading models is the Rayleigh fading channel model. It is a reasonable model reflecting the characteristics of the wireless propagation environment with a large number of scatterers. According to the Central Limit Theorem, each channel tap follows a Gaussian distribution with a zero mean if

there are sufficient scatters and no dominant line-of-sight path. In this case, the received signal is a superposition of multipath signals.

Unlike the flat Rayleigh fading channel widely used in the literature, the channels used in most of the simulations in this thesis are based on selective-fading channel models. Selective-fading channels mainly refer to frequency-selective and time-selective fading channels caused by time and frequency dispersion in the transmission process. Some specific parameters can be used to characterize the selective channel, such as the multipath delay spread for describing the frequency-selective fading and the Doppler spread for describing the time-selective fading.

As a result of time dispersion caused by multipath, the channel response changes with frequency. Coherence bandwidth is the frequency range during which the channel properties remain similar. When the coherence bandwidth of the channel is larger than the signal bandwidth, the transmitted signal is subject to flat fading. On the contrary, frequency-selective fading occurs if the coherence bandwidth of the channel is smaller than the signal bandwidth. The delay spread is inversely proportional to the coherence bandwidth. Therefore, the larger the delay spread is, the smaller the coherence bandwidth is, and for large bandwidth signals the channel becomes more frequency selective.

Moreover, coherence time is the parameter that describes the rate of channel variation. Depending on the value of the coherence time with respect to the symbol period, the signal experiences fast or slow fading. If the coherence time is shorter than the symbol period, the signal will undergo fast fading. The Doppler spread and coherence time are inversely related. Thus, when the coherence time is small, the Doppler spread is large, and the channel varies rapidly.

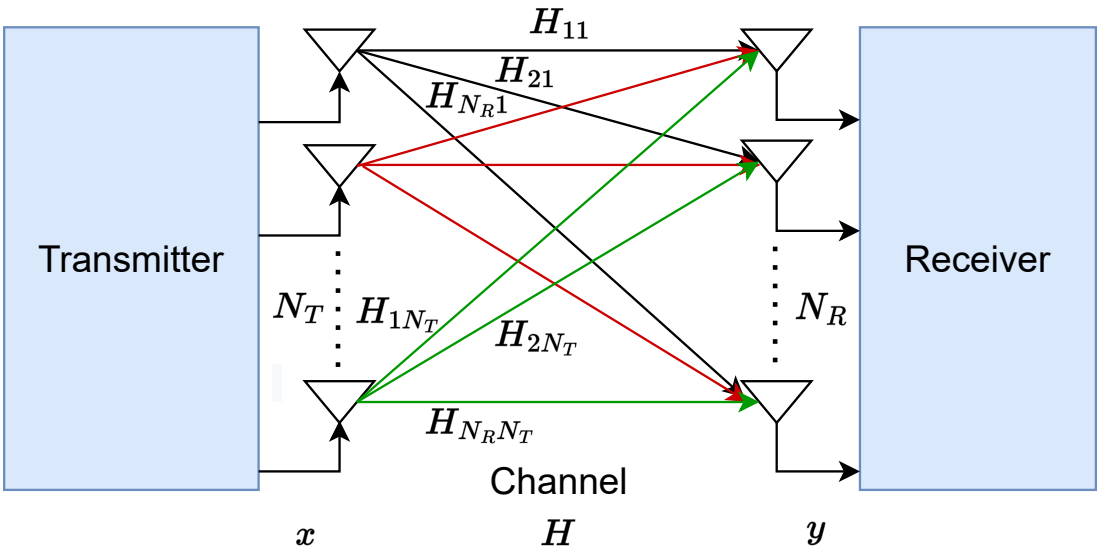


Figure 2.3: Block diagram of a $N_R \times N_T$ MIMO system.

2.2 Overview of Massive MIMO-OFDM Systems

2.2.1 MIMO Systems and Channel Models

Since the third generation (3G) wireless communication networks, one of the breakthrough technical advances at the physical layer (PHY) is the emergence of MIMO transceivers, which offer diversity gain and spatial multiplexing gain [19]. By using multiple antennas in the transmitter and the receiver, MIMO is able to improve the spectral efficiency, the system throughput, and the link reliability without increasing the system bandwidth and antenna transmit power.

In the MIMO system shown in Fig. 2.3, a multi-antenna transmitter sends multiple data streams simultaneously, and the transmit signals travel through wireless channels. Then, the receiver with multiple receive antennas gets the signal vectors filtered by channels and decodes them into the original information. Here we assume there are N_T transmit antennas and N_R receive antennas and the $\boldsymbol{H}$ is a Rayleigh-fading channel. Correspondingly, the elements of the channel matrix $\boldsymbol{H} \in \mathbb{C}^{N_R \times N_T}$ follow an

independent and identically distributed (i.i.d.) complex Gaussian distribution with a zero mean, i.e., $h_{ij} \sim \mathcal{CN}\left(0, \sigma_h^2\right)$. However, this uncorrelated MIMO channel model is not always applicable. Considering the spatial correlation of antennas in the statistical model of MIMO channels, the MIMO channel matrix $\boldsymbol{H}$ can be further expressed as:

$$\boldsymbol{H}_{sc} = \boldsymbol{R}_R^{1/2} \boldsymbol{H} \boldsymbol{R}_T^{1/2} \tag{2.5}$$

where $\boldsymbol{R}_R$ and $\boldsymbol{R}_T$ are the correlation matrices for the receive antennas and transmit antennas, respectively.

2.2.2 Massive MIMO-OFDM Systems

2.2.2.1 MIMO-OFDM Systems

As discussed in Subsection 2.1.2, realistic channels are often frequency selective rather than flat fading. Advanced equalization techniques and OFDM are the possible solutions to frequency-selective fading in MIMO systems. However, in a frequency-selective channel environment, the implementation of MIMO requires complex channel equalization schemes, which increases the complexity of the receiver significantly. As a multi-carrier modulation technique, OFDM divides a single radio channel into multiple narrowband flat-fading subchannels, which mitigates the impact of multipath channel environment and frequency-selective fading. Moreover, the Inter Symbol Interference (ISI) caused by multipath can be effectively eliminated by inserting a cyclic prefix (CP) [20]. Therefore, the use of OFDM can address the challenges of applying MIMO techniques under frequency-selective channels. The system model and mathematical description of OFDM can be found in Section 3.2.

Combining the advantages of MIMO in improving system capacity and OFDM in resisting frequency-selective fading, MIMO-OFDM is an effective way to meet the requirements of modern mobile communication systems, which has already attracted a lot of attention [21]-[22]. Like many other wireless communication standards, the fourth generation Long-Term Evolution (LTE) also employs MIMO-OFDM as one of the essential schemes [23]. For LTE, Single-carrier Frequency-Division Multiple Access (SC-FDMA) is used for uplink transmissions, and Orthogonal Frequency Division Multiple Access (OFDMA) is only used in the downlink. However, the overall system design becomes complex if the air interface is based on multiple waveforms. Considering the need for a single waveform and the scalability for diverse services, the fifth generation (5G) new radio (NR) employs OFDMA in both uplink and downlink [24]. Thus, it is practically important to further investigate MIMO-OFDM systems.

2.2.2.2 Massive MIMO

Since conventional MIMO base stations (BSs) only use a small number of antennas, the performance gain can not be fully exploited [25]. A massive MIMO system is a communications scheme that utilizes a large number (typically tens or hundreds, even thousands in the future) of antenna elements at base stations. In practice, the transmit antennas can be co-located in one transmitter or distributed in many terminals. Massive MIMO can further improve the energy efficiency, spectral efficiency, and robustness of LTE communication systems, which is considered as one of the key technologies of 5G. In addition, massive MIMO can use low-cost components like low-power amplifiers instead of expensive high-power ones, because each antenna only needs to be assigned a small fraction of the total transmit power [26].

Even if massive MIMO provides many advantages mentioned above, its large-scale antenna arrays introduce a series of new challenges for signal processing, such as beam precoding, multi-terminal synchronization, and pilot contamination. In this thesis, we mainly focus on the channel estimation and signal detection of massive MIMO-OFDM systems, and the relevant technical challenges are listed as follows:

1. *Channel Modeling*: Accurate performance evaluation of massive MIMO requires realistic channel models. There are additional characteristics of the channel to consider when using massive MIMO instead of conventional MIMO. However, conventional MIMO channel models like the widely-used WINNER II [27] fail to capture nearfield and non-stationary effects. Moreover, most prior channel models lack the features for time evolution and full three-dimensional (3D) propagation modeling which are important for massive MIMO channel modeling [28]. In this thesis, we develop a realistic channel generator based on an open-source channel simulator, QuaDRiGa [29], which includes full 3D propagation modeling. Furthermore, most of existing massive MIMO receivers focus on using the uniform linear arrays [7]. In Chapters 4 and 5, we utilize the rectangular array in our simulations, which also needs to be considered for massive MIMO systems.

2. *Channel Estimation*: Channel estimation plays a key role in wireless systems, especially in massive MIMO systems. To fully realize the potential of large-scale antenna arrays and obtain the capacity gain, acquiring the complete channel state information (CSI) at base stations is essential. However, the accurate estimation of CSI for massive MIMO channels is challenging [30], as the cost of acquiring CSI rises in proportion to the number of antennas. For example, in practice,

orthogonal pilot sequences are widely used to estimate wireless channels due to their satisfying performance. However, the overhead of orthogonal pilots increases with the total number of terminal antennas in the uplink, which leads to low spectral efficiency for massive MIMO-OFDM systems.

3. *Signal detection*: Some conventional detectors can perform well under small-scale MIMO systems and simple channel models. However, more advanced schemes are required for massive multiuser (MU) MIMO detection under real-world channels with frequency and spatial correlations. Due to a large number of antennas and radio frequency chains, the complexity of symbol detectors increased rapidly in a massive MIMO uplink receiver, which makes the design of detection algorithms more challenging. Thus, the research to find the efficient massive MIMO detection algorithm with optimal performance and low complexity has gained much attention during the past decade [31]. Moreover, the robustness to imperfect CSI and hardware impairments like low-precision analog-to-digital converters (ADCs) should also be considered for massive MIMO detectors.

2.2.3 Signal detection for MIMO-OFDM Systems

To provide a relevant background for massive MIMO detection, this subsection reviews several typical detectors as well as an iterative MIMO detector which is the basis of one deep learning (DL)-based benchmark approach in Chapter 4.

For an uplink MU-MIMO system with M user terminals with a single antenna and a base station equipped with N antennas, the relationship between the transmitted symbol vector $\boldsymbol{x} \in \mathbb{C}^M$ and the received vector $\boldsymbol{y} \in \mathbb{C}^N$ is given by:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{2.6}$$

where $\boldsymbol{n} \in \mathbb{C}^N$ is the additive white Gaussian noise with zero mean and variance N_0.

It is well-known that the maximum likelihood detection (MLD) can achieve the optimal performance. MLD finds the estimate $\widehat{\boldsymbol{x}}_{MLD}$ that minimizes the Euclidean distance between $\boldsymbol{H}\boldsymbol{x}$ and $\boldsymbol{y}$ [18], which leads to the integer least-squares problem:

$$\widehat{\boldsymbol{x}}_{MLD} = \arg\min_{x \in \chi^M} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 \tag{2.7}$$

where χ is the finite set of constellation points. We assume all user terminals use the same constellation set, and every symbol in χ is randomly chosen by the users with a uniform probability. The MLD algorithm requires exponential complexity in the number of users M, which is prohibitive for large-scale MIMO systems. In practice, low-complexity approaches are often used as alternatives.

As the basic detection method, the linear equalization like zero forcing (ZF) can be viewed as a relaxed solution for the MLD problem (2.7) [32]. Linear equalization-based detectors can be summarized as $\boldsymbol{x} = \boldsymbol{G}\boldsymbol{y}$, and the ZF and linear minimum mean-square error (LMMSE) detectors are given by [16]:

$$\widehat{\boldsymbol{x}}_{ZF} = \boldsymbol{G}_{ZF}\boldsymbol{y} = \left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\boldsymbol{H}^H\boldsymbol{y} \tag{2.8}$$

$$\widehat{\boldsymbol{x}}_{LMMSE} = \boldsymbol{G}_{LMMSE}\boldsymbol{y} = \left(\boldsymbol{H}^H\boldsymbol{H} + \sigma_n^2\boldsymbol{I}_M\right)^{-1}\boldsymbol{H}^H\boldsymbol{y} \tag{2.9}$$

where $\boldsymbol{G}_{ZF}$ and $\boldsymbol{G}_{LMMSE}$ are the equalization matrices, and $\boldsymbol{I}_M$ is an $M \times M$ identity matrix. With the help of the noise information σ_n, the LMMSE detector outperforms the ZF detector, especially when the noise power is large.

It is clear that both the ZF (2.8) and LMMSE (2.9) detectors require a matrix inversion operation in the equalization process. Due to the large-size channel matrix in massive MIMO systems, the computational complexity of matrix inversion is consider-

ably high. Thus, a series of approximate matrix inversion (AMI)-based algorithms are proposed for massive MIMO detection, which can avoid the matrix inversion operation. The conjugate gradient (CG) detector is an efficient AMI-based scheme that can achieve a near-LMMSE performance iteratively. To derive the CG-based detector, we first rewrite the LMMSE algorithm (2.9) as follows:

$$\widehat{\boldsymbol{x}} = \left(\boldsymbol{H}^H\boldsymbol{H} + \sigma_n^2\boldsymbol{I}\right)^{-1}\boldsymbol{H}^H\boldsymbol{y} = \boldsymbol{A}^{-1}\tilde{\boldsymbol{y}} \tag{2.10}$$

where $\boldsymbol{A} = \boldsymbol{H}^H\boldsymbol{H} + \sigma_n^2\boldsymbol{I}$ is the MMSE detection matrix, and $\tilde{\boldsymbol{y}} = \boldsymbol{H}^H\boldsymbol{y}$ is the matched filter vector. According to [33], the transmitted symbol vector at the $(i+1)$th iteration can be estimated as:

$$\widehat{\boldsymbol{x}}^{(i+1)} = \widehat{\boldsymbol{x}}^{(i)} + \alpha^{(i)}\boldsymbol{p}^{(i)} \tag{2.11}$$

where $\alpha^{(i)}$ is a scalar step size, and $\boldsymbol{p}^{(i)}$ is the conjugate direction with respect to $\boldsymbol{A}$, i.e.,

$$\left(\boldsymbol{p}^{(i)}\right)^H\boldsymbol{A}\boldsymbol{p}^{(j)} = 0, \quad \text{for } i \neq j \tag{2.12}$$

Based on Equations (2.10)-(2.12), the CG-based MIMO detection algorithm [34] can be described as:

$$\begin{aligned}
\widehat{\boldsymbol{x}}^{(i)} &= \widehat{\boldsymbol{x}}^{(i-1)} + \frac{\left(\tilde{\boldsymbol{y}}^{(i-1)}\cdot\tilde{\boldsymbol{y}}^{(i-1)}\right)}{\left(\boldsymbol{A}\boldsymbol{p}^{(i-1)}\cdot\boldsymbol{p}^{(i-1)}\right)}\boldsymbol{p}^{(i-1)}, \\
\tilde{\boldsymbol{y}}^{(i)} &= \tilde{\boldsymbol{y}}^{(i-1)} - \frac{\left(\tilde{\boldsymbol{y}}^{(i-1)}\cdot\tilde{\boldsymbol{y}}^{(i-1)}\right)}{\left(\boldsymbol{A}\boldsymbol{p}^{(i-1)}\cdot\boldsymbol{p}^{(i-1)}\right)}\boldsymbol{A}\boldsymbol{p}^{(i-1)}, \\
\boldsymbol{p}^{(i)} &= \tilde{\boldsymbol{y}}^{(i)} + \frac{\left(\tilde{\boldsymbol{y}}^{(i)}\cdot\tilde{\boldsymbol{y}}^{(i)}\right)}{\left(\tilde{\boldsymbol{y}}^{(i-1)}\cdot\tilde{\boldsymbol{y}}^{(i-1)}\right)}\boldsymbol{p}^{(i-1)}.
\end{aligned} \tag{2.13}$$

To simplify the notations, the step size $\alpha^{(i)}$ in (2.11) and another step size $\beta^{(i)}$ can be used in the CG iterations (2.13), which are given by:

$$\alpha^{(i)} = \frac{\left(\tilde{\boldsymbol{y}}^{(i-1)} \cdot \tilde{\boldsymbol{y}}^{(i-1)}\right)}{\left(\boldsymbol{A}\boldsymbol{p}^{(i-1)} \cdot \boldsymbol{p}^{(i-1)}\right)} \tag{2.14a}$$

$$\beta^{(i)} = \frac{\left(\tilde{\boldsymbol{y}}^{(i)} \cdot \tilde{\boldsymbol{y}}^{(i)}\right)}{\left(\tilde{\boldsymbol{y}}^{(i-1)} \cdot \tilde{\boldsymbol{y}}^{(i-1)}\right)} \tag{2.14b}$$

As shown in [34], the performance of the CG algorithm tends to be unsatisfactory when the ratio between BS antennas and user antennas, i.e. N/M, becomes small.

As discussed in Subsection 2.2.2, the massive MIMO detector has been a hot research topic in recent years. Thus, there are many detection techniques proposed for massive MIMO systems with different advantages and limitations [35]-[36]. However, the rule of the best trade-off between performance and complexity as well as the robustness of massive MIMO detectors under realistic environments are still open issues, which are also some of the main focus of this thesis. The next section will introduce deep learning techniques and discuss how they can be utilized to improve wireless communication systems.

2.3 Overview of Deep Learning Algorithms

As mentioned in Section 1.1, future cellular networks are becoming more complex due to larger and larger amounts of data from new applications and services. Apart from the massive MIMO and millimeter wave (mmWave) technologies used in 5G NR, machine learning is becoming another enabling technology to guarantee the stringent requirements of the sixth generation (6G) communication networks [8]. Thus, the AI-native network will be one of the key features of 6G communication systems.

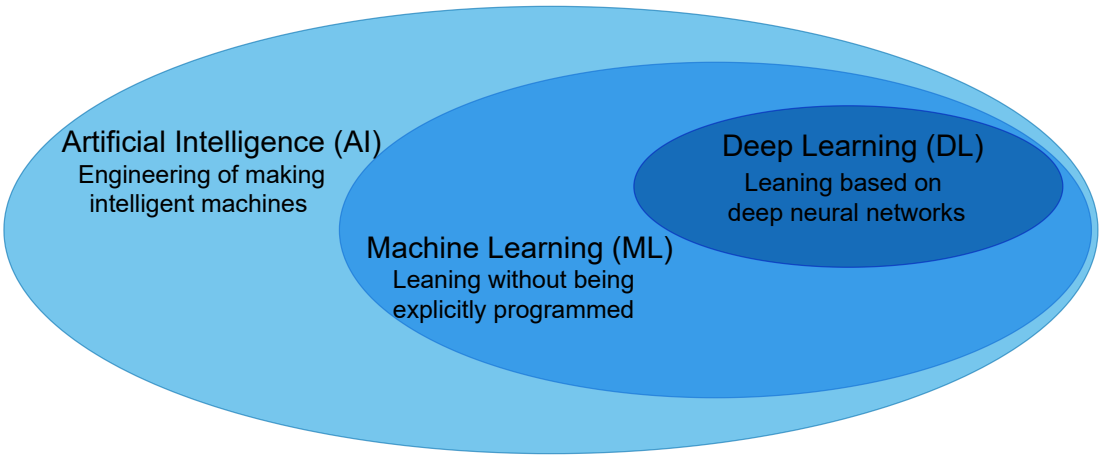


Figure 2.4: The relation between Deep Learning, Machine Learning, and Artificial Intelligence.

As a powerful instance among ML algorithms, deep learning (DL) is a promising technique for handling large data and enhancing future communication networks. In this section, we introduce the fundamentals of DL and then discuss the applications of DL techniques in wireless communications, especially in PHY communications.

2.3.1 Deep Learning

In general, AI is an umbrella term for the technologies aiming to teach machines how to think, react, learn and work in a similar way to human intelligence. As an essential branch of AI, ML algorithms can learn knowledge from data without explicit programming, which can be briefly categorized into supervised, unsupervised, and reinforcement learning. DL is a class of machine learning techniques that consists of multiple layers of information processing stages in hierarchical architectures [37]. Due to its data-driven nature, the DL model can automatically learn high-level features from raw data instead of manual feature extraction, which is a major benefit over conventional ML. The relation between DL, ML, and AI is illustrated in Fig. 2.4. This thesis mainly studies channel estimation and signal detection algorithms based on supervised deep learning.

2.3.1.1 Fundamentals of Deep Learning

Inspired by biological neural systems, DL is developed based on artificial neural networks (ANNs) with representation learning. The ANNs with more than one hidden layer can be called deep neural networks (DNNs). Note that the widely-used DNN is not the only architecture of DL. Other multilayer methods like deep random forests [38] can also be viewed as DL models. DNNs are generally used as a powerful function approximator to realize the mapping between input data and the desired output data, which can be interpreted in terms of the universal approximation theorem [39]. According to this theorem, if given a sufficient number of hidden neurons, a feedforward neural network with as few as one hidden layer can potentially approximate an arbitrary function.

Mathematically, a deep neural network can be characterized as a concatenation of a multitude of parameterized transformations as follows:

1. *Layers*: The basic architecture of NNs comprises an input layer, one or more hidden layers, and the output layer which finally transforms the data into the desired output. Depending on the connection modes among the neurons, the hidden layer can be a fully-connected layer, a convolutional layer, or other types of layers. The number of hidden layers (depth) and the number of neurons in different layers are essential hyper-parameters for the architecture of DNNs.

2. *Hidden Units*: The hidden unit is also called a neuron, which can be represented by a (usually non-linear) activation function. In a neuron in the hidden and output layers, the bias is added to the weighted input data, and the data is then processed by the activation function. Commonly used activation functions include the sigmoid:

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{2.15}$$

the Hyperbolic Tangent (tanh):

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.16}$$

and the Rectified Linear Unit (ReLU):

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2.17}$$

To better explain the principles of the training and inference processes of DNNs, we illustrate the forward propagation and backward propagation of a simple one-dimensional (1D) Convolutional Neural Network (CNN) in Fig. 2.5. The process in which the input vector $\boldsymbol{x}$ propagates to the hidden units at each layer and produces the final output $\boldsymbol{y}$ is called forward propagation. In the first hidden layer, the convolutional operation can be expressed as:

$$\boldsymbol{a}_1 = f(\boldsymbol{k}_1 * \boldsymbol{x}) \tag{2.18}$$

where $\boldsymbol{k}_1$ and $\boldsymbol{a}_1$ indicate the convolutional filter and the output of the first hidden layer, respectively. $f()$ is the non-linear activation function. After processing by the next convolutional layer and the output layer, the final output $\boldsymbol{y}$ is obtained. The aim of training this CNN is to learn the best parameter set $\boldsymbol{k} = [\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3]$ which can minimize the error between the predicted output $\boldsymbol{y}$ and the true value $\tilde{\boldsymbol{y}}$. After the forward propagation of data, this error can be calculated by the mean squared error (MSE) loss function $L(\boldsymbol{k})$ as follows:

$$L(\boldsymbol{k}) = \min_{\boldsymbol{k}} \frac{1}{S} \sum_{s=1}^{S} \|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|^2 \tag{2.19}$$

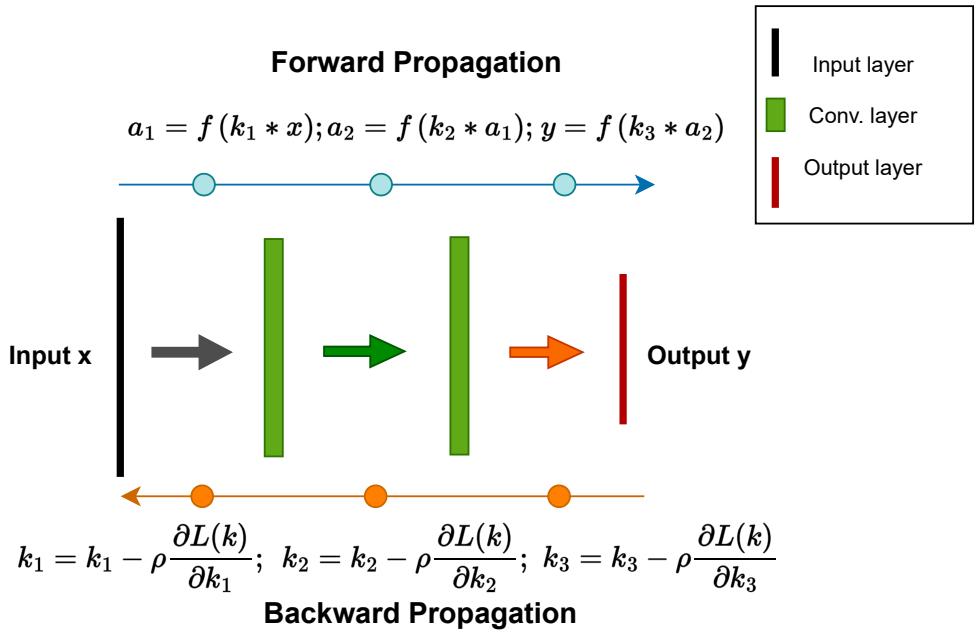


Figure 2.5: The illustration of the forward propagation and backward propagation processes of a CNN.

where S is the total number of training samples. $L(\boldsymbol{k})$ can be minimized by the backward propagation using a gradient descent algorithm. The backward propagation algorithm can help to compute the gradient by allowing the information from the loss function to flow backward through the network [40]. During backward propagation, the weight of the last layer, i.e., $\boldsymbol{k}_3$, can be updated by:

$$\boldsymbol{k}_3 = \boldsymbol{k}_3 - \rho\frac{\partial L(\boldsymbol{k})}{\partial \boldsymbol{k}_3} \tag{2.20}$$

where ρ is the learning rate. $\frac{\partial L(\boldsymbol{k})}{\partial \boldsymbol{k}_n}$ is the gradient of the loss function $L(\boldsymbol{k})$ over the weight of the nth layer, which can be computed following the chain rule [40]:

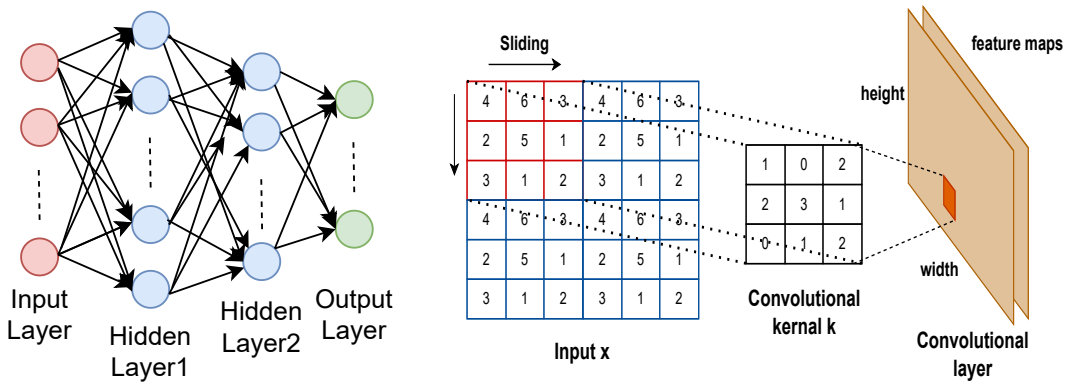
$$\frac{\partial L(\boldsymbol{k})}{\partial \boldsymbol{k}_n} = \frac{\partial L(\boldsymbol{k})}{\partial \boldsymbol{y}_N}\frac{\partial \boldsymbol{y}_N}{\partial \boldsymbol{x}_N}\cdots\frac{\partial \boldsymbol{x}_{n+1}}{\partial \boldsymbol{y}_n}\frac{\partial \boldsymbol{y}_n}{\partial \boldsymbol{x}_n}\frac{\partial \boldsymbol{x}_n}{\partial \boldsymbol{k}_n} \tag{2.21}$$

where N is the number of layers. The training process of a NN is comprised of forward propagation and backward propagation, as shown in Fig. 2.5. The update process in (2.20) will repeat until it converges.

2.3.1.2 Categories of Deep Learning Algorithms

There exist many DL architectures with different types of layers used for various tasks, such as the fully-connected DNN, CNN, Recurrent Neural Network (RNN), Autoencoder (AE), and Generative Adversarial Network (GAN). This subsection briefly introduces several network architectures relevant to the DL-based channel estimation and signal detection schemes in Chapters 3 and 5. Except for these generic data-driven DNN architectures, we propose two model-driven DL architectures with specialized network skeletons, which will be described in Chapters 4 and 6.

Multilayer Perceptron Multilayer Perceptron (MLP), also called fully-connected DNN, is a basic DL architecture. An MLP with two hidden layers is illustrated in Fig. 2.6 (a), where each node indicates a computational unit called a neuron. In a neuron, each output of the neurons in the previous layer is multiplied by a corresponding weight. In other words, each neuron is connected to adjacent layers, and the weights represent the strength of the connections. Then, the bias is added to the weighted sum of the input, and the data is finally processed by an activation function like Equations (2.15)-(2.17). MLPs can be trained efficiently to learn the best parameter set to minimize the loss function via the stochastic gradient descent optimizer or its variants. However, due to its fully-connected structure, MLP requires to learn a substantial number of weights, which leads to high complexity, especially for high-dimensional input data.



(a) Structure of an MLP. **(b)** Operating principle of a 2D convolutional layer.

Figure 2.6: The typical structure of MLP and CNN.

Convolutional Neural Network Based on the ideas of sparse interaction, parameter sharing, and equivariant representations, CNN is proposed to reduce the number of trainable parameters and improve the performance of traditional DNNs in high-dimensional data processing tasks [40]. With special layers for convolution, CNNs are particularly useful in processing the data with grid-like structures, such as two-dimensional (2D) images. As shown in Fig. 2.6 (b), a typical 2D convolutional layer takes the matrices with one or more channels as the input. Unlike full connections between the layers of MLPs, each locally connected kernel in the convolutional layer slides across the whole input matrix and forms a feature map. In addition to its remarkable achievements in image processing, CNN has started to show great potential in signal estimation and recovery problems, which will be demonstrated in the following chapters.

2.3.2 Deep Learning Applications in PHY Communications

DL techniques provide a new option for designing advanced wireless communication solutions. The standardization body International Telecommunications Union (ITU) has proposed the initiative for involving DL in future cellular networks and suggested various application cases [41]. Thus, DL-based schemes have been applied to different layers of wireless communications. For example, hierarchical Clustering is used for anomaly detection in mobile wireless networks [42]. In [43], the authors propose a data-driven resource management framework for ultra-dense small cells.

In recent years, DL has garnered growing interest in signal processing applications in the physical layer, e.g., interference management [44], modulation recognition [45], and channel coding [46]. Since the receiver design, including channel estimation and signal detection, is the focus of this thesis, we mainly introduce the relevant DL-based schemes in the rest of this subsection.

For realistic channels that may be non-linear and non-stationary, the analytical form of the optimal estimator is difficult to be derived [8]. On the contrary, DL-based estimators can be optimized for complicated channel environments through training and do not restrict to any specific channel statistics. In [47], an autoencoder-based channel estimation solution is proposed for vehicle-to-everything systems. To better adapt to time and frequency selective channels, the authors in [48] introduce online learning into a DNN-based channel estimator for single-input single-output (SISO)-OFDM systems. For mmWave MIMO systems, a DL-based scheme regards the channel matrix as an image and denoises it [49]. Moreover, a novel DL-based framework is developed for uplink channel estimation in hybrid analog-digital massive MIMO systems [50].

Compared with the SISO-OFDM or single-carrier MIMO systems in the abovementioned papers, the channel estimation task in massive MIMO-OFDM systems is more challenging due to high-dimensional channel matrices. By viewing the channel matrix as an image and exploiting the spatial and frequency correlation, CNNs have strong potential to achieve competitive estimation performance, as demonstrated in [51].

In signal detection, the aim of classical detection theories is to find the best estimate of received symbols, while data-driven deep learning tries to learn the optimal detection algorithm. In [9], [52], the authors propose an AE as an end-to-end transceiver system instead of considering the traditional block-based architecture. Unlike symbol-by-symbol detection schemes, an RNN-based sequence detection algorithm is proposed for molecular communication [53]. However, these detection solutions do not consider multi-carrier modulation schemes like the widely-used OFDM. With the assumption of perfect CSI, the authors in [54] use the extreme learning machine to detect OFDM symbols.

Existing data-driven DL-based detectors have a series of limitations, such as the unexplainable structure, which prevent their implementation in some realistic communication systems. In addition, training the generic DNNs of these schemes requires a long training time as well as a very large training dataset, which is currently scarce in wireless communications [55]. These findings motivate some researchers and us to fuse expert knowledge with data-driven DNNs, resulting in model-driven DL-based receivers. With the aid of traditional algorithms, a model-aided DNN-based receiver can handle non-linear power amplifiers in MIMO-OFDM systems [56]. There is another way to realize the combination of model-based algorithms and DL techniques, i.e., deep unfolding. Instead of finding the optimal values for step sizes like the α and β in (2.14) analytically, the CG-based network in [57] learns to optimize these step sizes via deep unfolding tools.

2.4 Summary

In this chapter, we first introduce large-scale and small-scale fading in wireless mobile communication channels, which is particularly useful for understanding the frequency-selective and time-variant channels used in Chapters 3-5. Next, for the complicated massive MIMO-OFDM systems considered in Chapters 4-6, the benefits, technical challenges, and relevant signal detectors are discussed. Then, the training and inference process of DNNs is illustrated with an example of a CNN architecture which will be used in Chapters 3 and 5. Finally, some DL-based solutions in PHY communications are briefly described.

Due to the constant growth of mobile devices in addition to complicated scenarios and unknown channel models in future communications, deep learning has become a promising approach for PHY communications. However, DL is not yet well-investigated in massive MIMO systems, and there exist some open issues to be solved [31]. For instance, most prior works are based on generic data-driven DL architectures, which heavily rely on a huge amount of data and time-consuming training. Unfortunately, in the wireless communications domain, the amount of training data is not comparable to the huge data sets used for core DL applications like computer vision [8]. Moreover, from the perspective of the no free lunch theorem, we should design specialized DL algorithms to perform well on a specific task rather than seek a universal or absolute best learning algorithm [40].

Motivated by these findings, we investigate how to integrate state-of-the-art DL techniques with communications domain knowledge efficiently. The core of this thesis is to develop model-driven DL-based receivers for massive MIMO-OFDM systems and realistic channels, which can achieve a reasonable trade-off between performance and complexity. In Chapters 3 and 5, several model-based algorithms are used to aid the proposed DNN-based channel estimators and signal detectors, resulting in the model-

aided DL architectures. On the other hand, Chapters 4 and 6 use another way to realize the idea of model-driven DL, i.e., unfolding the iterations of specialized algorithm architectures as the layers of neural networks and training their parameters via DL tools.

Chapter 3

Deep Neural Network-based Channel Estimation and Signal Detection for OFDM Systems

3.1 Introduction

Orthogonal frequency-division multiplexing (OFDM) is a multi-carrier modulation scheme that has been widely used in modern digital communications like Long-Term Evolution (LTE) and the fifth generation (5G) cellular systems to address frequency-selective fading in wireless channels. For wideband mobile communication systems, the radio channel is usually frequency-selective and time-variant [58]. For proper recovery of transmitted symbols, channel state information (CSI) should be estimated by the use of pilots which are known to both transmitter and receiver. Generally, traditional channel estimation methods include least squares (LS) [59], minimum mean-square error (MMSE), and their optimized versions based on different interpolation schemes [60], [61]. Even if the channel estimation is optimal, the noise and other distortions such as hardware impairments cannot be ignored at the receiver. Thus, a robust detection algorithm is also necessary for OFDM systems. Considering the high computational complexity of the maximum likelihood (ML) detector, the zero-forcing (ZF) detector and the MMSE detector [16] are usually popular choices.

Along with the increasing applications of advanced communication systems and the corresponding complex channel models, conventional channel estimation and data detection algorithms mentioned above form a computational bottleneck in real-time implementation [2]. The resurgence of artificial intelligence (AI) techniques especially deep learning (DL) offers an alternative option that is possibly superior to traditional ideas with respect to performance [4].

3.1.1 Literature Review

Deep Learning is a class of machine learning techniques that consists of multiple layers of information processing stages in hierarchical architectures [37]. The power of DL has been shown in many challenging applications like computer vision and speech processing [62], [63]. Recently, DL has garnered growing interest in the mobile communication networking domain [5]. For example, DL has been applied to different function blocks in the physical layer (PHY), e.g. modulation recognition [45], CSI feedback [64], and a polar codes decoder [65]. Moreover, DL-based channel estimation for vehicle-to-everything systems has been proposed in [47], which introduces the autoencoder (AE) into the conventional data-pilot aided process. In [49], the channel matrix of beamspace mmWave massive multiple-input multiple-output (MIMO) systems is used as an image and then estimated by a denoising neural network.

For signal detection tasks, there are several different DL-based schemes, such as sequence detection and symbol-by-symbol detection. Firstly, [53] presents the sequence detection algorithm for molecular communication based on a recurrent neural network (RNN). Secondly, the authors in [9], [52] propose an AE-based system to represent the entire end-to-end communication system as an alternative to designing specific modules in each traditional function block. The AE-based end-to-end system is further extended to MIMO tasks under the simple Rayleigh fading channel in [66]. Also, there

are some other studies in DL-based MIMO detection based on simple channel models such as [67]. However, these abovementioned detection algorithms do not consider realistic channel models or multi-carrier modulation schemes like OFDM. In [54], the authors utilize the DL model to detect OFDM signals but assume the CSI is perfectly known, which is unrealistic in practice.

Moreover, channel estimation and other blocks at the OFDM receiver are considered as a black box and embedded in one fully connected deep neural network (FC-DNN) in [68]. Even if its performance is competitive, the FC-DNN requires a long training time together with a huge amount of training data to train a large number of parameters. In addition, this end-to-end receiver is not designed to calculate the channel time-frequency response explicitly. Hence, it is not efficient for some applications that rely on accurate CSI like massive MIMO systems. Beyond this, an long short-term memory-based OFDM receiver with a linear channel estimator is presented in [69]. Due to the limited expressive and generalization ability of linear neural networks without activation functions, the channel estimator in [69] needs to use pilots in all subcarriers before each data symbol is sent, which consumes significant spectrum resources.

3.1.2 Contributions

In order to solve the issues mentioned above, in this chapter, we propose a hybrid OFDM receiver named RecNet (an abbreviation of receiver network) which integrates a channel estimation neural network (CE-NN) and a signal detection neural network (SD-NN). The CE module in our system can estimate the time-frequency response of fast time-variant and frequency-selective channel models. In the 2D (time-frequency) channel matrix, only the channel response at the pilot positions is known. The remainder of the channel response is estimated by the proposed convolutional neural network (CNN)-based CE-NN. The output matrix of CE-NN includes the complete

CSI that is used for the detection of received signals. Then the original bits are recovered by the fully connected SD-NN which jointly conducts detection and demodulation. Both the proposed CE-NN and SD-NN are trained offline with the aid of traditional communication algorithms, which can be referred to as model-driven deep learning. The main contributions of this chapter are presented as follows:

1. By exploiting the parameter sharing and learning capacity of CNN, a low-complexity channel estimator, CE-NN, is designed for 2D channel estimation. Compared with the end-to-end DL-based receivers like [9], it can provide explicit CSI for the equalizer to obtain further performance improvement. Moreover, due to the robustness of CE-NN to the small number of pilots, our design of lattice-type pilots has high flexibility depending on how fast the channel changes in the time and frequency domain. This approach can provide a good trade-off between estimation accuracy and the number of pilots used.

2. With the help of the channel information obtained by the CE-NN and the initialization aided by traditional ZF equalization, the SD-NN in RecNet is trained to converge much faster and requires fewer training data than the data-driven solutions in the literature. In addition, to detect more data bits with one neural network, we develop a series of data preprocessing and optimization schemes to compensate for the performance reduction caused by the bigger output layer without increasing the model complexity. As a result, with lower overall complexity, RecNet achieves better performance than the state-of-the-art OFDM receiver in [68]. Finally, according to our simulation results, the robustness of the proposed OFDM receiver is demonstrated both in terms of SNR and length of the cyclic prefix (CP).

It is worth mentioning that a state-of-the-art NN-based OFDM receiver in [70] also includes a channel estimation network. However, the network architecture we used and the ways we utilize the estimated CSI are different. Our system achieves better performance and faster convergence speed by using CSI in the receiver, while the performance gain in [70] is mainly from the CSI-aided precoder.

The rest of this chapter is organized as follows. Section 3.2 provides the background and mathematical description of the OFDM system we considered in this chapter. In Section 3.3, we present the architecture of the proposed channel estimator and the model details of CE-NN. Section 3.4 introduces the technical challenges, data preprocessing, and model tuning schemes of SD-NN in detail. In Section 3.5, simulation results and analysis of complexity are shown to demonstrate the performance and robustness of our low-complexity algorithms when compared to other channel estimation and signal detection methods. Finally, Section 3.6 provides the conclusions to this chapter.

3.2 System Model

Generally, there are three elements: the transmitter, wireless channels, and the receiver in an OFDM communication system. The architecture of the OFDM system with the proposed DL-based receiver is provided in Fig. 3.1.

The OFDM transmitter modulates the message bits into a sequence of Quadrature Amplitude Modulation (QAM) symbols. Then these symbols will be subsequently converted into N parallel streams. Each of the N symbols from the serial-to-parallel (S/P) conversion is mapped to a different subcarrier. After the inverse fast Fourier transform (IFFT) operation, the cyclic prefix (CP) that should be longer than the

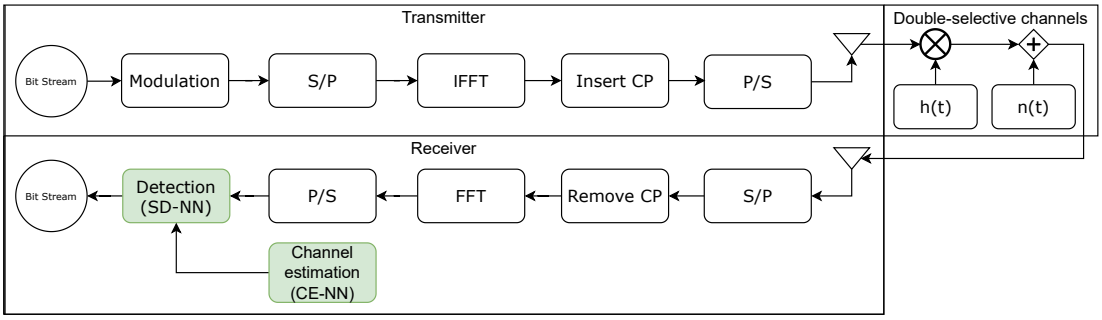


Figure 3.1: The architecture of OFDM system with the proposed DL-based receiver.

maximum delay spread of the multipath channel is added in the front of OFDM symbols to mitigate intersymbol interference (ISI). Then, the time-domain signal with CP is converted back to serial form and transmitted to propagate through the multipath channel $\{\boldsymbol{h}(l)\}_{l=0}^{L-1}$ with L paths. The received signal in the time domain is:

$$\widetilde{y}(l) = \widetilde{x}(l) \otimes h(l) + \widetilde{n}(l) \tag{3.1}$$

where $\otimes$ means circular convolution, and $\widetilde{x}(l)$, $h(l)$, and $\widetilde{n}(l)$ are the transmitted signal, the channel impulse response (CIR), and the additive white Gaussian noise (AWGN), respectively. After CP removal and the fast Fourier transform (FFT) operation, the received signal for the kth subcarrier can be expressed as:

$$y(k) = H(k)x(k) + n(k) \tag{3.2}$$

where $x(k)$, $H(k)$, and $n(k)$ represent the transmitted signal, the channel response, and the noise in the frequency domain, separately.

The wireless channel considered in this chapter is not only frequency-selective but also time-varying (i.e. doubly-selective). Thus, estimating the channel response for the entire OFDM frame that is composed of the OFDM symbols in different time slots is

necessary. In our system, the coherence interval of the fast-varying channel model is equivalent to one OFDM symbol, that is, the channel response is varying between each OFDM symbol. So the time-frequency response of the channels can be considered as a 2D matrix. For the jth time slot and kth subcarrier, the transmit-receive relationship in (3.2) can be further represented as:

$$y_j(k) = H_j(k)x_j(k) + n_j(k) \tag{3.3}$$

where $H_j(k)$ is the (k, j) element of $\boldsymbol{H} \in \mathbb{C}^{K\times T}$, and $\boldsymbol{H}$ is a matrix containing the channel response of all K subcarriers and T time slots in a frame. The output of the channel estimation module (CE-NN), $\boldsymbol{H}_{est}$ which contains the complete estimated values of $\boldsymbol{H}$is then used for the initialization of the detection network (SD-NN). These two neural networks are highlighted in green color in Fig. 3.1.

3.3 Channel Estimation Neural Network (CE-NN)

3.3.1 CNN-based Channel Estimation

The CNN was first presented in [71] three decades ago and has become one of the most popular neural networks. Due to the underlying ideas of parameter sharing, equivariant representations, and sparse interactions, CNN performs better than traditional DNNs in high-dimensional data processing tasks, especially image processing. The essential motivation for introducing a CNN into channel estimation is the parameter reduction capability of CNN, which is important for achieving a low-complexity channel estimation network. Furthermore, the CNN is suitable to process the time-frequency channel response that has a similar 2D structure to images.

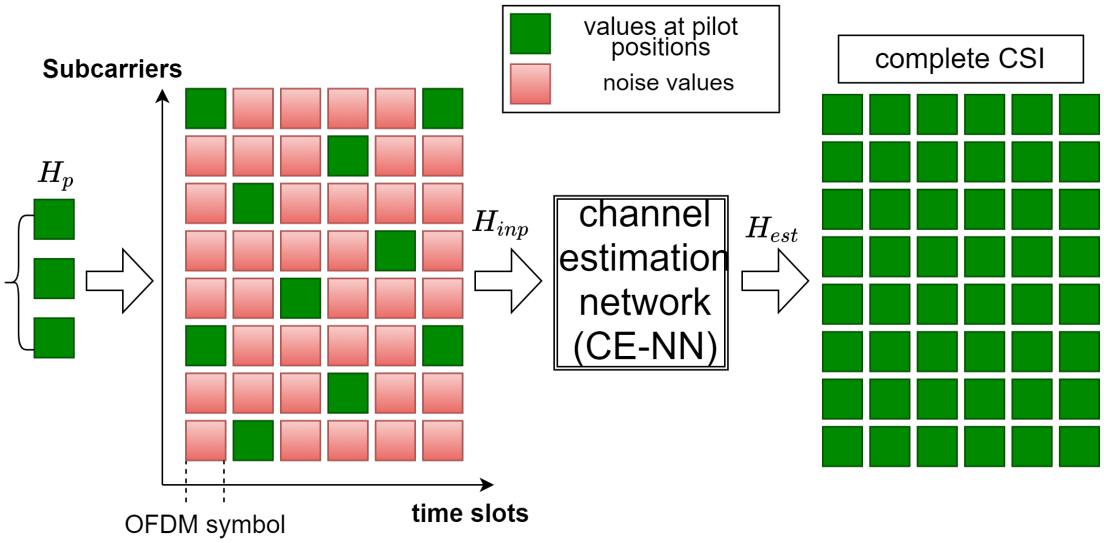


Figure 3.2: The proposed CNN-based channel estimation scheme.

For pilot-based channel estimation methods, the pilot pattern is the basis for the subsequent work. Basically, there are three different types of pilot patterns: block-type, comb-type, and lattice-type. Among them, the lattice-type pilot is very suitable for the estimation of frequency-selective and fast-fading channels. For the one-dimensional (1D) DNN-based channel estimation [69], the channel in each time slot is estimated by a full OFDM symbol with block-type pilots in the frequency domain. Compared with this scheme which needs several full pilot symbols and 1D-NNs to estimate the channel response for the whole OFDM frame, the proposed combination of lattice-type pilots and a 2D-CNN can significantly decrease the required number of pilots and the computational complexity for channel estimation. It is because the lattice-type pilot we developed is inserted along both time and frequency axes, and the corresponding CNN-based CE-NN can jointly estimate the channel response in all time slots of the whole OFDM frame.

The flowchart of the proposed CNN-based channel estimation is shown in Fig. 3.2. Unlike pure data-driven schemes, the proposed CE-NN is initialized by a conventional

channel estimation to improve the convergence and performance. First, the initial channel response at the pilot positions is calculated by the use of the LS method:

$$\widehat{H}_p = y_p / x_p \tag{3.4}$$

where x_p indicates the known pilot values and y_p is the corresponding received values after FFT. To get the complete channel matrix including the channel response at both pilot and data subcarriers, 2D interpolation corresponding to the lattice-type pilots should be applied, i.e., the $\widehat{H}_p$ in (3.4) are interpolated in time and frequency axes to obtain the initial channel values at non-pilot positions. Here we adopt the bicubic interpolation method customized to our 2D lattice-type pilot pattern. It can generate a smoother interpolated surface than other 2D interpolation schemes, such as bilinear interpolation or nearest-neighbor interpolation. The interpolated values of the bicubic method can be calculated as:

$$f(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} \mathrm{a}_{i,j} x^i y^j \tag{3.5}$$

where $\mathrm{a}_{i,j}$ denotes the interpolation coefficient, and x^i, y^j are the coordinate values of along two axes. There are many super-resolution techniques for image processing, such as interpolation-based, sparse-coding-based [72] and learning-based schemes [73]. If consider the estimated 2D channel response $\widehat{H}_p$ at the pilot positions as a low-resolution image, then both the bicubic-based interpolation and the proposed CE-NN in Fig. 3.2 can be viewed as the super-resolution reconstruction. To further enhance the estimation performance for doubly-selective channels, we combine traditional channel estimation based on interpolation with the CNN-based CE-NN. More precisely, the

received pilots are first preprocessed by the LS algorithm (3.4) and bicubic-based interpolation (3.5) for initialization. Then the coarse 2D channel matrix $\boldsymbol{H}_{intp}$ with interpolated errors is fed to CE-NN to yield denoised values of channel response in all time slots and subcarriers, i. e. $\boldsymbol{H}_{est}$, as shown in Fig. 3.2.

3.3.2 Model and Training Specification of CE-NN

In our OFDM system, each OFDM symbol that has $K = 64$ subcarriers is transmitted in a time slot, and each OFDM frame has $T = 14$ symbols. Note that the real and imaginary parts of the complex-valued $\boldsymbol{H}_{intp}$ need to be separated and then input into CE-NN. Thus, for each OFDM frame, the input size of CE-NN is $(64, 14, 2)$. $(64, 14, 2)$ is also the size of $\boldsymbol{H}_{est}$, i.e. the output of CE-NN. Since the input and output of each layer in CE-NN have the same size, there is not a specific input layer in CE-NN. It just consists of three convolutional layers. In general, a convolutional layer includes multiple convolutional filters. Each of them processes data only for its receptive field and then the summation is calculated by a sliding window. The first layer takes the matrix $\boldsymbol{H}_{intp}$ as input and produces the output as follows:

$$\boldsymbol{S}_1 = f\left(\boldsymbol{W}_1 * \boldsymbol{H}_{intp} + \boldsymbol{c}_1\right) \tag{3.6}$$

where $\boldsymbol{W}_1$, $\boldsymbol{c}_1$, and $\boldsymbol{S}_1$ indicate the convolutional filters, the biases, and the output of the first layer, respectively. The second and third layers will do the same with the new inputs $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. The operator $*$ means the convolution operation, and $f(\cdot)$ denotes the activation function. A linear activation function is used in the third layer to reconstruct the final output, and the first two layers utilize the Rectified Linear Unit (ReLU) as their activation function:

$$F_r(a) = \max(0, a) \tag{3.7}$$

Table 3.1: The layers and parameters of CE-NN.

Layer	Filter number, size	Activation function	Weight initializer	Trainable parameters
Convolutional layer 1	$64, 9 \times 9$	ReLU	*He norm* [74]	5248
Convolutional layer 2	$32, 1 \times 1$	ReLU	*He norm*	2080
Output layer	$1, 5 \times 5$	Linear	*Tru norm*, $stddev = 0.001$	801

The input matrix of CE-NN is the sample set of $\boldsymbol{H}_{intp}$, and the corresponding output is the set of $\boldsymbol{H}_{est}$.

The summaries of CE-NN's parameters, such as the number of filters and the number of trainable parameters, are all included in Table 3.1. The first layer has 64 filters of size 9×9, and the second layer has 32 filters of size 1×1, which are suggested by [73]. The last layer uses only one filter of size 5×5 to reconstruct the channel. The increase of filter number and size in each layer can decrease the error of CE-NN, but the choice of network scale is always a trade-off between performance and complexity. *He norm* denotes the normal distribution-based weight initialization scheme proposed in [74], which shows the state-of-the-art performance especially in the hidden layers with ReLU activation. *Tru norm* is the truncated normal distribution-based weight initializer, and *stddev* means the standard deviation. According to our experiments, for the last layer with linear activation function, *Tru norm* with $stddev = 0.001$ outperforms *He norm* in our situation. Moreover, the total number of trainable parameters for a convolutional layer in Table 3.1 is calculated in the following function:

$$N_p = C_i K^2 F_n + F_n \tag{3.8}$$

where C_i denotes the input channel, K is kernel size, and F_n is the number of filters used in the convolutional layer. The computational complexity of CE-NN will be calculated together with SD-NN as a complete OFDM receiver in Section 5.

The Adam optimizer is employed with the default settings in [75], except the initial learning rate is 0.002 with a batch size of 256. Like some other hyper-parameters in Table 3.1, these two values are also empirical choices. To be more precise, we find this initial learning rate can increase the convergence rate and also improve the model performance on the test dataset. Correspondingly, the batch size is also increased from 128 to 256, which can help the proposed CE-NN converges more stably within a short convergence time. In the training phase, we generate the CIR based on the WINNER II channel model [27], which is convolved with the random transmitted signals in the time domain, as in equation (3.1). As discussed in Subsection 3.3, the received pilot signals are preprocessed to get the initial estimates of channels, i.e. $\boldsymbol{H}_{intp}$, which are then packaged into the different training and testing datasets in the frequency domain. After this, the original CIR is converted into the CFR used as the training label of CE-NN. The trainable parameters are updated according to the mean squared error (MSE) loss function, and this network only needs to be trained by 100 epochs. Finally, the output of CE-NN is reshaped as a new complex-valued matrix $\widehat{\boldsymbol{H}}$ that is then used to improve the performance and convergence speed of the detection network SD-NN, which will be described in the following section.

3.4 Signal Detection Neural Network (SD-NN)

In the era of big data, the deep neural network (DNN) has been proven to be a well-performing universal approximator [39] due to its ability to handle large data and learn features automatically. With the help of a huge amount of labeled data, a data-driven DNN is usually the first choice in many fields, especially in natural language processing and computer vision. However, training a data-driven network requires sufficient labeled data and computing resources, both of which are rarely found in wireless communication devices. In addition, the extensive training time and high computational complexity of the pure data-driven DNN model make its hardware implementation in most of the physical layer applications very challenging. Inspired by the model-driven concept of deep learning [76] and its extension in communication [55], we propose an NN-based OFDM receiver, RecNet. The most crucial characteristics of RecNet are fast convergence, low complexity, and reduced numbers of pilots, which are achieved by the model-aided design and further model optimization of the CE-NN presented in Section 3 and SD-NN described in this section.

3.4.1 Technical Challenges

The explosion of advanced wireless applications, such as virtual reality and Internet of things, has propelled the development of wireless communication into 5G to achieve a thousand-fold capacity, millisecond latency, and massive connectivity [2]. As one way to meet these requirements, the modulation order has increased from 4-QAM in 3G to up to 256-QAM in 5G. Accordingly, the design of signal detectors corresponding to higher-order modulation becomes more challenging. For the DL-based OFDM receivers in [68] and [70], Quadrature Phase Shift Keying (QPSK) (or 4-QAM) is used for modulation. For the OFDM systems with 64 subcarriers in these papers, there

are 128 bits in one OFDM symbol that need to be detected. As an example, the output size of the neural network in [68] is only 16, which means the detection of each OFDM symbol needs eight independent NNs under 4-QAM. The ComNet proposed in [69] has a similar problem. Obviously, these kinds of solutions are not very efficient for high-order modulation schemes of 5G which would require tens of NNs just for one-symbol detection. Furthermore, the high overall system complexity makes these systems hard to be implemented on the chips like Field Programmable Gate Arrays (FPGA).

As well as the computational complexity, the convergence speed should also be considered as a critical performance indicator for NN-based detectors, which makes an efficient design of DL-based detection algorithms challenging. Without the help of explicit CSI and model-aided initialization, the training process of some DL-based detection solutions requires massive data and extensive time (e.g., spending a day for single training with thousands of epochs) to converge, e.g. [67], [68]. In conclusion, low complexity and fast convergence are essential for the implementation of DNN-based schemes in PHY layer communications.

3.4.2 The Architecture of RecNet

Fig. 3.3 shows the complete architecture of the proposed RecNet. Based on the domain knowledge of OFDM systems, equalization is the main way to mitigate channel distortion via introducing the estimated CSI into received signals. After the transmission process in (3.3), the complex received signal $\boldsymbol{y}$ is used as one input of the ZF equalizer, and another input $\widehat{\boldsymbol{H}}$ is the complex-valued version of the CE-NN's output $\boldsymbol{H}_{est}$. The ZF-based equalization function for the jth OFDM symbol can be expressed as:

$$\boldsymbol{x}_j^{init} = \boldsymbol{y}_j / \widehat{\boldsymbol{H}}_j \tag{3.9}$$

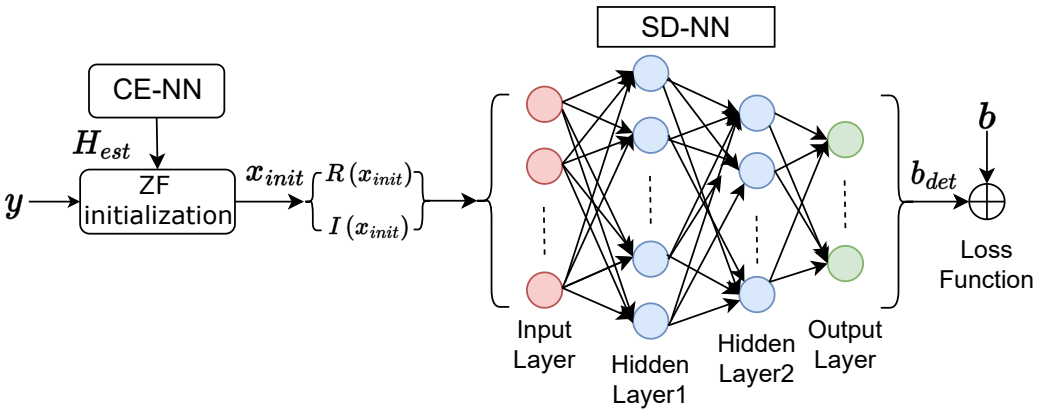


Figure 3.3: The architecture and data flow of the proposed RecNet.

With the help of estimated CSI, the ZF equalizer can maximally eliminate the ISI of the received symbols and provides a proper initial value $\boldsymbol{x}_{init}$, i.e. the input of the detection network SD-NN. Thus, the ZF equalization (3.9) is used as initialization for SD-NN in a model-aided manner, as shown in Fig. 3.3. However, the ZF equalizer can have a huge performance loss under low SNR conditions since it does not take the noise information into account. Hence, the detection network, SD-NN will take the output of ZF equalizer to denoise, demodulate and mitigate other non-linear distortions that are difficult to be formulated and processed by well-defined mathematical models. Note that the input of SD-NN, $\boldsymbol{x}_{init}$ needs to be separated in the real part $\Re\{\boldsymbol{x}_{init}\}$ and imaginary part $\Im\{\boldsymbol{x}_{init}\}$ first and then concatenated. Each OFDM symbol is composed of 64 subcarriers, so the input size of SD-NN is 128. In Fig. 3.3, the label $\boldsymbol{b}$ and loss function are only applicable to the training process.

As shown in Fig. 3.3, apart from the input layer, SD-NN includes two hidden layers and an output layer. These four layers can be described mathematically by the following equations:

$$\begin{aligned} I &= \boldsymbol{x}_{init} \\ \boldsymbol{h}_1 &= F_r\left(\boldsymbol{W}_1 \boldsymbol{x}_{init} + \boldsymbol{c}_1\right) \\ \boldsymbol{h}_2 &= F_r\left(\boldsymbol{W}_2 \boldsymbol{h}_1 + \boldsymbol{c}_2\right) \\ \boldsymbol{b}_{det} &= F_s\left(\boldsymbol{W}_3 \boldsymbol{h}_2 + \boldsymbol{c}_3\right) \end{aligned} \tag{3.10}$$

where I, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$ and $\boldsymbol{b}_{det}$ represent the outputs of these four layers respectively, and $\boldsymbol{W}_l$, $\boldsymbol{c}_l$ indicate the weights and bias of the SD-NN. $F_r(\cdot)$ is the ReLU activation in function (3.7) and $F_s(\cdot)$ denotes the sigmoid function as follows:

$$F_s(x) = \frac{1}{1+e^{-x}} \tag{3.11}$$

As the label of SD-NN is the original binary symbol $\boldsymbol{b}$, sigmoid activation is utilized in the output layer to map the output in the range (0, 1). Since the aim of the proposed SD-NN is to get the output vector $\boldsymbol{b}_{det}$ that is close to the training label $\boldsymbol{b}$, the trainable parameters $\theta = \{\boldsymbol{W}_l, \boldsymbol{c}_l\}_{l=1}^{3}$ are optimized through the following L_2 loss function:

$$L_2\left(\boldsymbol{b}; \boldsymbol{b}_{det}\left(\boldsymbol{x}_{init}, \theta\right)\right) = \min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \left\| \boldsymbol{b}^n - \boldsymbol{b}_{det}^n \right\|^2 \tag{3.12}$$

where N indicates the amount of training data, and $\boldsymbol{b}^n$, $\boldsymbol{b}_{det}^n$ are the supervision label and the network output that corresponds to the nth training sample, separately.

Table 3.2 illustrates the details of SD-NN's configuration, including the layer size, activation function, weight initializer, and the number of learnable parameters for each layer. As mentioned above, the size of the input layer is 128. There are 300, 150, and 128 neurons in the three fully connected (FC) layers, respectively. As shown in equation (3.10), ReLU is still the activation function of all hidden layers, and the sigmoid activation is adopted in the output layer. Weight initializer 1 and Weight initializer 2 are the different weight initializers used in the two different configurations

Table 3.2: The layers and parameters of SD-NN.

Layer	Layer size	Activation function	Weight initializer 1	Weight initializer 2	Trainable parameters
Input layer	128	...	...	...	0
Hidden layer 1	300	ReLU	$Tru\ norm$, $stddev = 0.01$	$Xav\ norm$	38700
Hidden layer 2	150	ReLU	$Tru\ norm$, $stddev = 0.1$	$Xav\ norm$	45150
Output layer	128	Sigmoid	$Tru\ norm$, $stddev = 0.05$	$Xav\ norm$	19328

which are designed for varying degrees of non-linear effects. $Tru\ norm$ and $Xav\ norm$ denote truncated and Xavier normal distributions, respectively. Further details of these two configurations will be discussed in the next subsection. Moreover, the formula to calculate the number of trainable parameters for the FC layers in Table 3.2 is given by:

$$M_p = N_i N_o + N_o \tag{3.13}$$

where N_i and N_o are the input and output sizes of each layer, respectively.

3.4.3 Data Preprocessing

As presented in Table 3.2, the model size of the SD-NN in this chapter is just slightly bigger than the one we proposed in [10], but its output size is increased four times from 32 to 128. With the help of the bigger output layer, one neural network can detect more bits, so the overall complexity of the receiver system is dramatically decreased. However, according to our simulation, if we simply increase the output size of the NN-based detectors without increasing the network size like layer width or model depth, the performance will noticably decrease at the same time. As discussed in Subsection

3.4.1, it is also why one network in [68] and [69] only detects the bits for 8 consecutive subcarriers. To compensate for the performance loss of the bigger output size of SD-NN without increasing the training time and model complexity, we develop a series of data preprocessing and model optimization schemes.

First, the analysis of the input data is essential for data preprocessing. As shown in Fig. 3, the input of SD-NN is $\boldsymbol{x}_{init}$. According to the scatter plots of the data in equation (3.9), the equalized signal $\boldsymbol{x}_{init}$ has several very large values (outliers) compared with the received signal $\boldsymbol{y}$, since there are some very small values of channel response in the matrix $\widehat{\boldsymbol{H}}$. For example, for the same training batch, the range of $\boldsymbol{x}_{init}$ is [-88.2, 97.4], but the range of $\boldsymbol{y}$ is only [-1.4, 1.57]. This makes $\boldsymbol{x}_{init}$ hard to be handled by the SD-NN with small initial weights. Thus, the normalization for such equalized signals is necessary to help SD-NN perform better and converge faster.

Table 3.3 compares the data range, mean and standard deviation of $\boldsymbol{x}_{init}$ with different normalization schemes. Even if the Min-Max normalization can scale the input data into (0, 1), which is similar to the range of the training labels, it is still not a good choice in our system. This is because it makes the difference between input samples become too small and it cannot handle the outliers like some very large values of $\boldsymbol{x}_{init}$. According to the histogram and the small standard deviation 0.0155, most values of $\boldsymbol{x}_{init}$ are very close to the mean 0.4752 after Min-Max normalization. Unlike the Min-Max method, the Z-score normalization can scale the input data into a proper range and maintain the distance between samples. The mathematical expression of Z-score normalization is:

$$F_z(x) = \frac{x-\mu}{\sigma} \tag{3.14}$$

Table 3.3: The range and distribution of the input data with different normalization.

	$\boldsymbol{x}_{init}$	with Min-Max	with Z-score
Range	$[-88.2, 97.4]$	$[0.00034, 0.99985]$	$[-30.65, 33.93]$
Mean	0.0255	0.4752	5.88×10^{-7}
Standard Deviation	2.8734	0.0155	0.9999

where μ and σ denote the mean and standard deviation of the data x, respectively. As shown in Table 3.3, $\boldsymbol{x}_{init}$ is approximately rescaled to be a standard normal distribution with mean zero and standard deviation one, which makes the random training and testing data that are equalized by the fast-varying channel response satisfy the independently and identically distributed (i.i.d) assumption.

3.4.4 Model Tuning

For the tuning of deep learning models, choosing the depth of the network and the width of each layer are the primary considerations. The successful implementation of neural networks with deeper structures which are often harder to train and optimize is an excellent achievement of machine learning techniques. However, on the basis of our experimental results, the deeper network is not always better for the OFDM detection task in this chapter, especially considering the trade-off between performance and complexity. With the same parameter settings, the SD-NN with 512, 256, and 128 neurons in 3 hidden layers only has a 5% lower average bit error rate (BER) but 2.45 times more trainable parameters than the one with 2 hidden layers in Table 3.2. Because the trade-off between computational complexity and performance is usually the main consideration of communication algorithms, using the smaller size neural network with proper training strategies may be a good compromise.

Table 3.4: The two different configurations of SD-NN.

	Weight initializer	Batch normalization	Initial learning rate	Learning rate decay strategy
Config1	Truncated normal	...	0.001	Step decay, every 20 epochs
Config2	Xavier normal	Before the output layer	0.002	Natural exp. decay

As mentioned in [37], the no free lunch theorem for machine learning implies that we must design our machine learning algorithms to perform well on a specific task rather than seek a universal learning algorithm or the absolute best learning algorithm. Coincidentally, the goal of the wireless signal detection algorithms is to get a good trade-off between performance and complexity in a specific radio environment. Therefore, to address the technical challenges in Subsection 3.4.1 and adapt to different levels of non-linear effects, we propose two different configurations of SD-NN. Table 3.2 and Table 3.4 give the differences between the first configuration (config1) and the second configuration (config2), including the weight initializers and the strategies of learning rate decay.

In the training phase, the proper initial values of weights can help DL models converge rapidly and find global minima. For the normal situation without non-linear effects, the weights of each layer are initialized by the truncated normal distribution with different standard deviations, as shown in Table 3.2. In this case, the standard deviations can be manually set to get a competitive performance based on the analysis of input data and the trials of training. When only a very few pilots are used in the pilot symbol, and the CP length is shorter than the multipath delay, bigger errors of channel estimation and non-linear effects will be introduced in the input data of SD-NN. In this situation, it is difficult to find empirically a suitable setting for weight initializers based on

truncated normal distribution. For the SD-NN with sigmoid activation in the output layer, the Xavier initializer [77] is utilized as the weight initializer of the config2, as it outperforms the He initializer. Additionally, we add a batch normalization layer before the output layer to further reduce the potential influence of bad initial weights.

As one of the most critical hyper-parameters of deep learning, learning rate (LR) should be considered first along with the optimizer in the optimization phase. Usually, the Adam optimizer only needs an initial value of LR like the default 0.001 [75], since it can adapt the LR as learning unfolds. While in our practice of SD-NN's training, the Adam with the learning rate decay outperforms that with fixed initial LR. As shown in Table 3.4, there are two different configurations of SD-NN in this chapter. For config1 without considering non-linear effects, we design a step decay with two different decay rates based on lots of experiments and analysis. The LR starts at 0.001 and attenuates every 20 epochs for the 100-epochs training process. The decay rate at the 20th, 60th, and 100th epochs is 0.8, and at the 40th and 80th is 0.5. Under the situation with non-linear effects such as short CP, it is difficult to empirically find a proper setting of step or polynomial LR decay. Therefore, we utilize the natural exponential decay for config2, which is smoother than the step decay and attenuates faster than the normal exponential decay. The initial LR and decay rate of this natural exponential decay are 0.002 and 0.8, separately. The batch size for the two configurations of SD-NN is 400.

3.5 Simulation results

In this section, several experiments have been conducted to demonstrate the performance and robustness of the 2 proposed neural networks for different lengths of CP and a range of SNRs. Simulation results of the conventional algorithms for channel estimation and signal detection are also involved in the following figures as the benchmark of the proposed scheme. Moreover, the comparison and analysis of the required number of pilots per symbol, convergence speed, and computational complexity between different OFDM receivers are also provided.

In our experiments, each OFDM frame consists of 14 OFDM symbols in the 14 time slots, which is consistent with the OFDM data frame of LTE and 5G new radio (NR). and each symbol contains 4 pilot subcarriers and 60 data subcarriers among 64 subcarriers. The length of the CP is 16 which is equal to the maximum delay spread of the multipath channel. The sizes of training, validation, and testing datasets of both CE-NN and SD-NN are 32000, 4000 and 4000, respectively. Both the original bits, wireless channel response, and AWGN are synthetic data. The simulation data of the wireless channel is generated based on the channel model named WINNER II [27] for the Non-Line of Sight (NLOS) scenario which is challenging for channel estimators. The multipath number is 24, and the carrier frequency is 2.6 GHz, which is similar to the channel configuration in [68] for a fair comparison. 16-QAM is used as the modulation scheme in our simulation. Moreover, the entire OFDM system, including the proposed CE-NN and SD-NN, is implemented in Python simulation platform with a TensorFlow backend. All the experiments including model training are run on a standard personal computer equipped with an Intel i7-8700 CPU and Nvidia GTX 1070 GPU. For the DL-based channel estimation and signal detection, MSE and BER are used as the performance metrics, respectively.

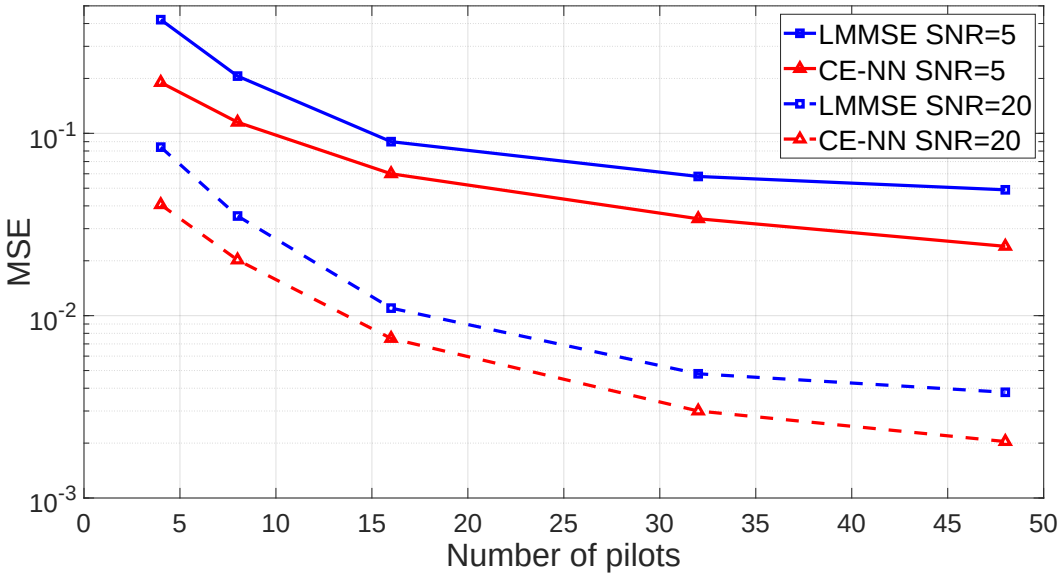


Figure 3.4: MSE curves of CE-NN and LMMSE versus the number of pilot signals.

3.5.1 Channel Estimation Performance

The CSI which is obtained by channel estimation algorithms not only can be used in equalization to improve the performance of signal detection, but also supports techniques used at the transmitting side like beamforming. Generally, the ideal MMSE estimator has the optimal performance with the help of perfectly-known channel correlation matrix, but that is unrealistic in practice. Therefore, a suboptimal version of MMSE, linear minimum mean-square error (LMMSE) which outperforms the LS estimator, is implemented as the baseline for the proposed CE-NN. More details of CE-NN's parameters are presented in Section 3 and Table 3.1.

Fig. 3.4 presents the results of both LMMSE and CE-NN at a low SNR (SNR = 5 dB) and a moderate-high SNR (SNR = 20 dB) for different number of pilots. Note that the pilot numbers along the X-axis in Fig. 3.4 mean the average number of pilots used in each OFDM symbol (each time slot). When the pilot number is smaller than 16, CE-NN has a much lower MSE than LMMSE for both SNRs, especially when

only 4 pilots are used in an OFDM symbol. For conventional CE methods, it is hard to obtain accurate estimation and interpolation without full knowledge of channel models. By contrast, the CE-NN can compensate for interpolation errors and minimize MSE further by learning the underlying structural features from the image-like 2D-interpolated channel matrices. Beyond 32 pilots, LMMSE starts to show a trend of saturation, but CE-NN still has the potential to reduce the MSE if more pilots are used. In addition, the FC-DNN in [68] requires up to 64 pilot subcarriers in a time slot, even if it does not explicitly estimate the CSI. We will compare the BER performance of the proposed RecNet and FC-DNN in the next subsection.

Moreover, in this figure, the gap of MSE performance between LMMSE and CE-NN is more significant at SNR = 5 dB than at SNR = 20 dB. For example, the CE-NN with 16 pilots has comparable performance to the LMMSE with 32 pilots when SNR = 5 dB, while CE-NN needs 24 pilots to achieve a similar MSE as the 32-pilot LMMSE when SNR = 20 dB. This is because low SNR has a huge influence on conventional estimation methods like LMMSE, but this effect is not that great for our CNN-based estimator. CE-NN can efficiently denoise the initial estimates that contain severe noise for low SNRs. In conclusion, the numerical results in Fig. 3.4 demonstrate the superior performance of CE-NN as well as its robustness against the small number of pilots and low SNRs.

3.5.2 Detection Performance

As shown in Fig. 3.3, the equalized received signal is the input of the SD-NN. We adopt the widely-used MMSE detection as the benchmark and also compare the proposed RecNet with the state-of-the-art detection neural network [68] using a similar configuration of the OFDM system but using higher-order modulation.

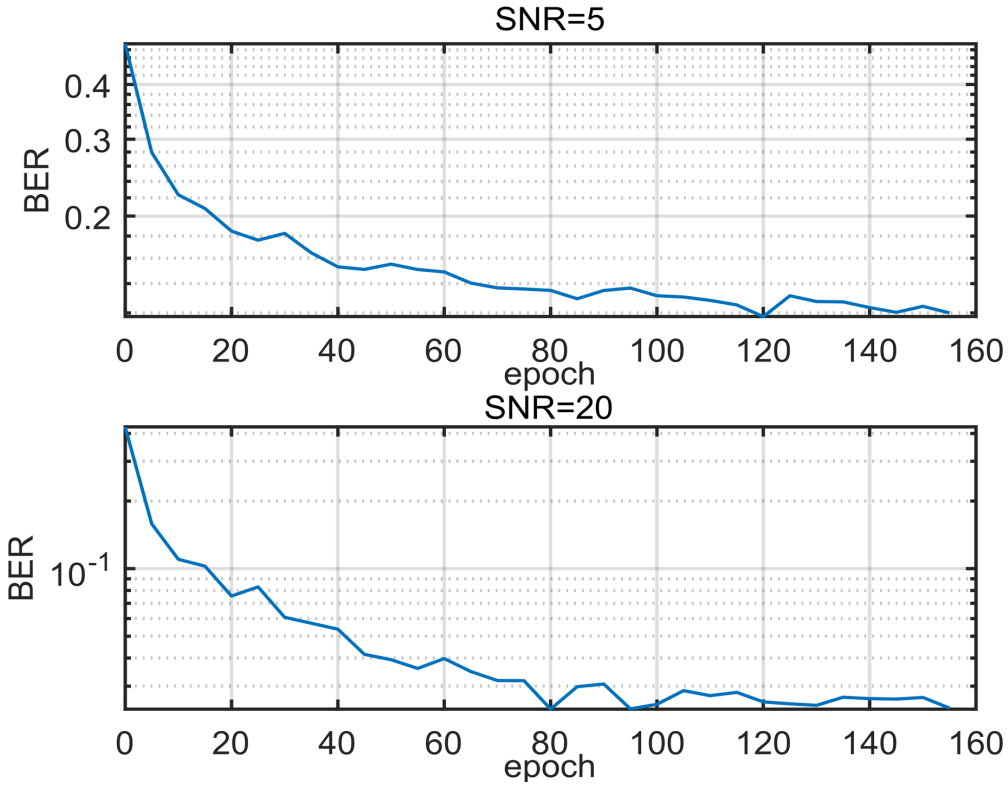


Figure 3.5: Convergence curves of SD-NN under SNR = 5 dB and SNR = 20 dB.

3.5.2.1 Convergence Property of SD-NN

The convergence curve of the proposed detection network SD-NN with config1 is illustrated in Fig. 3.5. There are only 56 pilots inserted into the whole frame that includes 14 OFDM symbols. To completely show the saturation trend of the BER curve, the training epochs are increased from 100 to 155. It is obvious that from the point of epoch = 80, the BERs under both SNRs start to show a trend of convergence. When the SNR equals 5 dB or 20 dB, the BER is 0.11 or 0.018 for 100 epochs, which are both lower than the corresponding BERs of the FC-DNN under our system configuration. With the help of the CE-NN in Section 3, the data preprocessing and model optimization strategies in Section 4, SD-NN demonstrates the capacity of faster

convergence compared to the FC-DNN [68] using 20000 training epochs. Furthermore, the performance of SD-NN is not fully converged within 100, or even 155 epochs. It still has the potential to reduce BER further with more training epochs, but the performance is less improved than before.

3.5.2.2 Performance analysis of RecNet

Fig. 3.6 compares the detection performance of SD-NN with config1 and config2, the traditional MMSE detection scheme and other DL-based approaches for doubly-selective channels. Note that the BER performance in this figure is obtained with estimated CSI rather than for the unrealistic perfect CSI case. For the OFDM systems with 16-QAM modulation and 64 subcarriers, there are 256 bits per transmitted symbol that need to be detected. In the legend of Fig. 3.6, we use abbreviations for the OFDM receivers with channel estimators and signal detectors as follows:

- LMMSE-MMSE: The combination of conventional LMMSE channel estimation and MMSE detection schemes. 16 pilots are used in each time slot.
- FC-DNN-64: The data-driven detection network [68] with 64 pilots in each time slot. The size of the output layer is changed from 16 to 32 to adapt to 16-QAM, which means the prediction of an OFDM symbol requires 8 independent FC-DNNs. The number of training epochs is 5000.
- FC-DNN-8: The FC-DNN with 8 pilots in each time slot. The output size is also 32, but the number of training epochs increases to 10000.
- RecNet1: The proposed RecNet consisting of CE-NN and the SD-NN with the config1 shown in Table 3.4. There are 56 pilots utilized for the channel estimation of 14 OFDM symbols in 14 time slots, which is equivalent to only 4 pilots in each time slot. The output size of RecNet is 128, which means the prediction of one OFDM symbol requires two RecNets. The number of training epochs is 100.

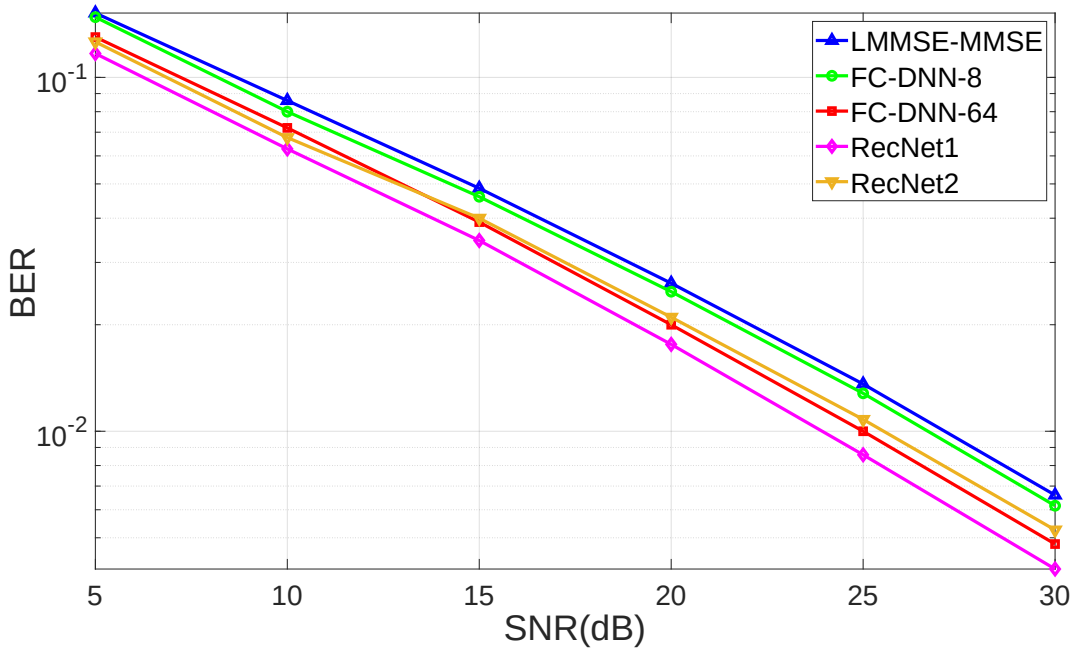


Figure 3.6: BER versus SNR curves of RecNet and other detection schemes.

- RecNet2: The proposed RecNet consisting of CE-NN and the SD-NN with the config2 designed to resist non-linear effects. The pilot number and output size remain the same as RecNet1, but the number of training epochs is increased to 120.

To demonstrate the robustness to different SNRs, the SD-NN in RecNet is trained under a fixed SNR (10 dB), whereas the results of RecNet in Fig. 3.6 are obtained by testing it over arbitrary SNRs. With the aim of compensating for the performance loss from the ZF-based initialization at low SNRs, we train the SD-NN under a relatively low SNR of 10 dB rather than a higher SNR. With the long-time allocated for training, the FC-DNN-8 slightly outperforms the traditional LMMSE-MMSE with 16 pilots. Compared to the FC-DNN-8, the FC-DNN-64 uses sufficient pilots to occupy all 64 subcarriers, thus achieving much lower BER at the cost of lower spectral efficiency. Even if it only uses 4 pilots in an OFDM symbol, the proposed RecNet1 still has better performance than the FC-DNN-64 and FC-DNN-8 over all SNRs.

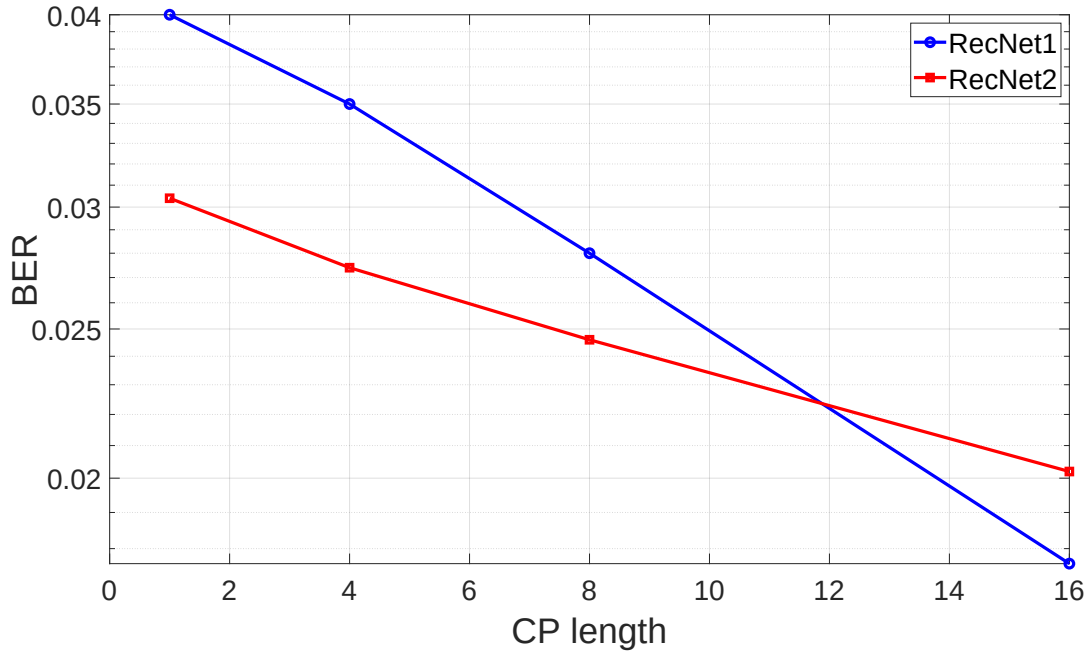


Figure 3.7: BER curves of RecNet1 and RecNet2 in terms of CP length.

Moreover, the FC-DNN-64 [68] requires 5000 epochs to achieve the BER in Fig 3.6, while RecNet1 starts to converge within 100 epochs, which is only 1/50 of the number of epochs required for the FC-DNN-64. With the help of the proposed CE-NN in Section 3, the model-aided architecture in Fig. 3.3, and the model tuning in Subsections 3.4.3-3.4.4, RecNet achieves competitive performance, pilot saving and fast convergence. Compared with the RecNet1 which needs fine-tuning of the weight initializers and learning rate based on a mass of experimental trials, RecNet2 performs slightly worse in Fig. 3.6 without the non-linear effects.

Generally, the CP is necessary for OFDM systems to eliminate the ISI. However, the long CP leads to extra costs and reduces spectrum efficiency. Fig. 3.7 provides the BER performance of RecNet1 and RecNet2 for different lengths of the CP. Both the RecNet1 and RecNet2 are trained at 10 dB SNR and tested at 20 dB SNR. When the length of CP equals 1, 4, or 8 which is shorter than the maximum multipath delay 16, RecNet2 shows a lower BER than RecNet1. However, RecNet1 performs better than

RecNet2 when the CP length is sufficient (CP= 16). This is because the RecNet1 is well-tuned under the linear case, but its performance degrades under the case with non-linear effects caused by the severe ISI. In contrast, the config2, including the different designs of initializer, learning rate and batch normalization layer, makes the RecNet2 more robust to shorter CP. In particular, even when the CP length is only 1, RecNet2 still works well, and its performance degradation is not significant compared to that with 16 CP. In summary, RecNet1 has better performance under the linear case with sufficient CP in Fig. 3.6, while RecNet2 has better robustness to non-linear effects like the insufficient length of CP in Fig. 3.7. These numerical results demonstrate the necessity to develop different configurations of the proposed SD-NN for varying levels of nonlinearity.

3.5.3 Complexity analysis

The performance, robustness, and convergence speed of the proposed RecNet have been demonstrated by our simulation results and analysis. In addition to these key performance indicators (KPIs), computational complexity is also an important evaluation index of deep learning models and data detection algorithms. It determines the operating speed and energy efficiency of the hardware implementation of these algorithms. Table 3.5 compares the KPIs of the proposed RecNet and other above-mentioned OFDM receivers in detail. The RecNet in this chapter has two different configurations, so there are two values of epoch number that correspond to RecNet1 and RecNet2 separately. Compared with both FC-DNNs, our RecNet requires less training data, much fewer pilots and training epochs to converge into the lower BER. Without considering expert knowledge, FC-DNN [68] uses a generic DNN architecture and fully random initialization, thus requiring a large training dataset and a long training time to converge.

Table 3.5: The comparison of the pilot number, epoch number and computational complexity for RecNet and competing OFDM receivers.

	Batch size	Size of training dataset	Pilot number per symbol	Epoch number of convergence	Complexity in MFlops per OFDM frame
Initial RecNet [10]	400	32000	4	100	16.1
RecNet	400	32000	4	100/120	5.7
FC-DNN-64 [68]	1000	50000	64	5000	64.5
FC-DNN-8	1000	50000	8	10000	64.5
LMMSE-MMSE	...	...	16	...	22.3

Directly comparing complexity is not easy, especially for the time complexity affected by complicated factors like simulation platforms and hardware implementation. To visually compare the complexity, we calculate the number of floating-point operations (FLOPs). For fully-connected DNNs like FC-DNN and SD-NN, the computational complexity of a single forward pass is $O\left(\sum_{k=1}^{L-1} n_i^k n_o^k\right)$, where L, n_i and n_o are the layer number of networks, input and output sizes of each layer, respectively. As mentioned in Subsection 3.5.2, the output size of FC-DNN is only set to 32 for an acceptable performance, which means four times the number of FC-DNNs are required compared with the proposed SD-NN. Note that the complexity in Table 3.5 is for the detection of 14 symbols in the entire OFDM frame rather than only one OFDM symbol. With the help of the CSI estimated by CE-NN and the model-aided initialization, the initial version of RecNet [10] has a smaller size and lower overall complexity than FC-DNN. In this chapter, a series of data preprocessing and model tuning strategies have been developed to further reduce the complexity of the detection subnetwork SD-NN without the loss of performance.

3.6 Summary

This chapter proposes a novel deep learning-based scheme and corresponding design methodologies for channel estimation and signal detection in OFDM systems. This OFDM receiver is divided into two low-complexity NNs based on domain knowledge in communications. First, a CNN is designed to estimate the doubly-selective channel by viewing it as a 2D image. Simulation results show that this channel estimator can efficiently refine conventional interpolated-based channel estimation and offer a competitive accuracy, especially when only a few pilots are used. Due to the flexible trade-off between the number of pilots and estimation performance, the proposed OFDM receiver has the potential to adapt to different channel models and various scenarios in the 5G era.

Based on the experiments and data analysis, several strategies of data preprocessing and network tuning are designed to reduce the complexity and improve the performance of the fully-connected NN detector. With the help of these optimization methods and the CSI obtained from our CNN, the receiver system requires much fewer training epochs than prior data-driven solutions. Due to the fast convergence rate and small training dataset size, this system can be trained rapidly to adapt to uncertain distortions during online deployment. Besides, we develop two different model configurations for varying degrees of non-linear effects. Our experiments with a short CP and a small number of pilots show the robustness of the proposed receiver RecNet. Finally, the computational complexity of RecNet and other OFDM receivers is compared to demonstrate its low complexity.

In summary, the results in this chapter have demonstrated the superiority of model-aided DNN-based receivers compared with pure data-driven schemes. Recently, the usage of expert knowledge and model-aided schemes in DL-based channel estimation and signal detection have been further developed, such as the MIMO-OFDM receiver

with non-linear power amplifiers [56]. However, the existing model-aided schemes, including our RecNet, are designed based on generic DNN models, so they are still limited by the nature of data-driven deep learning, such as uninterpretable architectures, large numbers of learnable parameters, and data-hungry training. These problems become especially serious when the data dimension is very large, like in large-scale MIMO-OFDM systems. Therefore, in the next chapter, we will investigate alternative model-driven deep learning schemes for massive MIMO-OFDM detection.

Chapter 4

A Deep Unfolding Network for Massive MU-MIMO-OFDM Detection

4.1 Introduction

Multiuser (MU) multiple-input multiple-output (MIMO) is a key technology for fifth-generation (5G) wireless communication systems. Massive MU-MIMO equips the base station (BS) with a large number of antenna elements that can serve many user terminals in the same frequency band [78]. This key enabler of 5G and beyond promises higher spectral efficiency and reliability at the cost of higher computational complexity. On the receiver side, transmitted symbols with interference and noise superpose, and the detection algorithms are supposed to separate these signals. As the number of antennas grows, the complexity of data detection increases exponentially. Thus, the detection task of massive MU-MIMO systems is very challenging.

Over the past few decades, many methods have been proposed for MIMO detection. Some non-linear detectors, such as Maximum likelihood (ML), have optimal performance with high complexity. On the contrary, sub-optimal linear detectors are less complex, like zero forcing (ZF) and the minimum mean squared error (MMSE). Based on the assumption of independent identically distributed (i.i.d.) Gaussian matrix, approximate message passing (AMP) with near-optimal performance is proposed in [79]. However, matrix inversion exhibits high computational complexity being one of the most complex operations in linear and simple non-linear MIMO detectors. For

massive MIMO systems, this problem becomes more severe as the dimension of the channel matrix increases [80]. Approximate matrix inversion (AMI)-based algorithms have been designed to reduce complexity by avoiding computing the matrix inverse. For instance, Gauss-Seidel and conjugate gradient (CG) methods can be found in the literature [81], [82].

4.1.1 Literature Review

The power of deep learning (DL) has been widely shown in the upper layers of wireless communication systems. Within several years, DL is beginning to gain momentum in signal processing applications like detection in the physical layer [5]. An autoencoder (AE)-based end-to-end detection in [9] views the whole communication system as a black box. Besides, the works in [10], [68] which redesign some of the function blocks in OFDM receivers via deep neural networks (DNN) and convolutional neural networks (CNN), are easier to be optimized compared with black-box designs for the entire transmitter and receiver. However, these two schemes still need a huge dataset and a long time to be trained since they use generic DNNs with huge numbers of trainable parameters.

Recently, MIMO detection based on model-driven DL has gained popularity due to its interpretable architecture and fast convergence. As a powerful instance, deep unfolding (DU) unfolds the iterations of model-based algorithms into a layer-wise structure analogous to a neural network and introduces learnable parameters [83]. One of the earliest DU-based MIMO detectors, DetNet, is presented in [67], [84]. DetNet provides competitive performance on a simple Gaussian channel. However, the number of its trainable parameters is still dependent on the number of antennas, which is not friendly for massive MIMO systems. For 4-Quadrature Amplitude Modulation (QAM), an alternating direction method of multipliers (ADMM)-based network [85] outperforms DetNet. However, the similar projection functions in [84] and [85] are designed only

for low-order modulations. To relax this limitation, a multilevel projection is proposed for a projected gradient descent based network in [86]. In addition, the works in [87], [88], [89] unfold different types of iterative AMP algorithms to detect the signals from MIMO channels. In [57], a hardware-friendly network, LcgNet is developed based on CG. Whereas the performance of AMI-based algorithms like CG greatly depends upon the settings of communication systems. Unlike long-term evolution (LTE), MU-MIMO-OFDM is the main scheme for uplink in 5G communications, which makes the design of detection algorithms challenging. However, most existing works only evaluate their networks in very simple settings (e.g., low-order modulation, single-carrier communication, point-to-point MIMO scenario, i.i.d. Gaussian channels). As in [90], DL-based schemes should be evaluated on realistic rather than simple channel models to avoid misleading conclusions for MIMO detection performance. Hence, MM-Net [90] introduces online learning and provides superior performance under real-world channels. The price paid is retraining for each new channel realization during online testing stages, which leads to a considerable increase in parameter numbers, training overhead, and latency. Motivated by the issues discussed above, we are interested in whether an offline detector can perform robustly in massive MU-MIMO-OFDM systems with realistic correlated channels.

4.1.2 Contributions

As mentioned above, DL-based receivers have shown promising results on wireless communication systems with simple settings. However, degraded performance or high complexity limits their universality in massive multiuser MIMO-OFDM systems, especially with high user load and realistic channels. In this chapter, we propose a novel deep unfolding-based iterative detection algorithm for massive MU-MIMO-OFDM systems, called MMO-Net. Our model-driven algorithm fuses over-relaxed ADMM (OR-ADMM) architecture with DU tools, which combines the power of

domain knowledge and data. For simple channel models like i.i.d. Gaussian channels, only one set of parameters is sufficient for all channel realizations. For the realistic 3GPP-3D channels, different sets of parameters are learned to handle the severe spatial-frequency correlations. Numerical results demonstrate that, with similar or lower complexity, the proposed MMO-Net outperforms traditional and state-of-the-art DL-based detection algorithms in massive MU-MIMO-OFDM systems. According to the simulation results, MMO-Net has superior robustness to the full range of user load and different channel models including the real-world 3GPP-3D channels. Moreover, due to the small number of layers and trainable parameters, this network can be trained to converge rapidly.

The main contributions of this chapter can be summarized as follows:

1. According to [91], with the optimal step size and relaxation parameter, OR-ADMM can outperform the classical and accelerated ADMMs in terms of convergence and performance. Thus, we first derive a MIMO detector based on the general OR-ADMM architecture. However, there is only a fixed step size and a relaxation parameter for all iterations in conventional OR-ADMM algorithms, which seriously restricts their flexibility. It is also difficult for traditional approaches to find the optimal set of algorithm parameters for different iterations as well as the different subcarriers in multi-carrier systems like OFDM. To solve this problem, we unfold the iterations of OR-ADMM in a layer-wise neural network and jointly learn its parameters explicitly for different subcarriers from the training data.

2. To better adapt to realistic correlated channels, we add two different trainable step sizes in $\boldsymbol{x}$- and $\boldsymbol{u}$-updates to increase the model flexibility. As a result, our MMO-Net with moderate flexibility converges faster than ADMM-Net [85] and has fewer learnable parameters than DetNet [84].

3. We analyze different non-linear operators and design a specific differentiable non-linear projection to ensure the whole model can be optimized by gradient descent optimizers. For high-order modulations like 16-QAM, this multilevel projection helps MMO-Net outperform the state-of-the-art ADMM-Net. Furthermore, with the same maximum and minimum as the constellation set χ, this projection can also perform as a proper constraint of the iterative values to improve performance and convergence.

4. We implement and adapt three state-of-the-art DU-based detection networks [57], [85], [88] into the massive MU-MIMO-OFDM system with high user load and real-world channels considered by few prior works. Correspondingly, extensive simulations and comparisons are conducted to provide more insights into model-driven DL-based detection tasks. To further reduce MMO-Net's complexity, we design an efficient implementation scheme sharing parameters between different subcarriers in the frequency domain.

The remainder of this chapter is organized as follows. Section 4.2 first introduces the structure of MU-MIMO-OFDM systems. Next, our ADMM architecture for MIMO detection is described in Section 4.3. Section 4.4 discusses the proposed MMO-Net including over-relaxed design, the choice of extra trainable parameters, the non-linear projection, and the comparison with ADMM-Net. Then the simulation results are presented to demonstrate the performance and robustness of MMO-Net in Section 4.5. Finally, Section 4.6 gives the conclusion of this chapter.

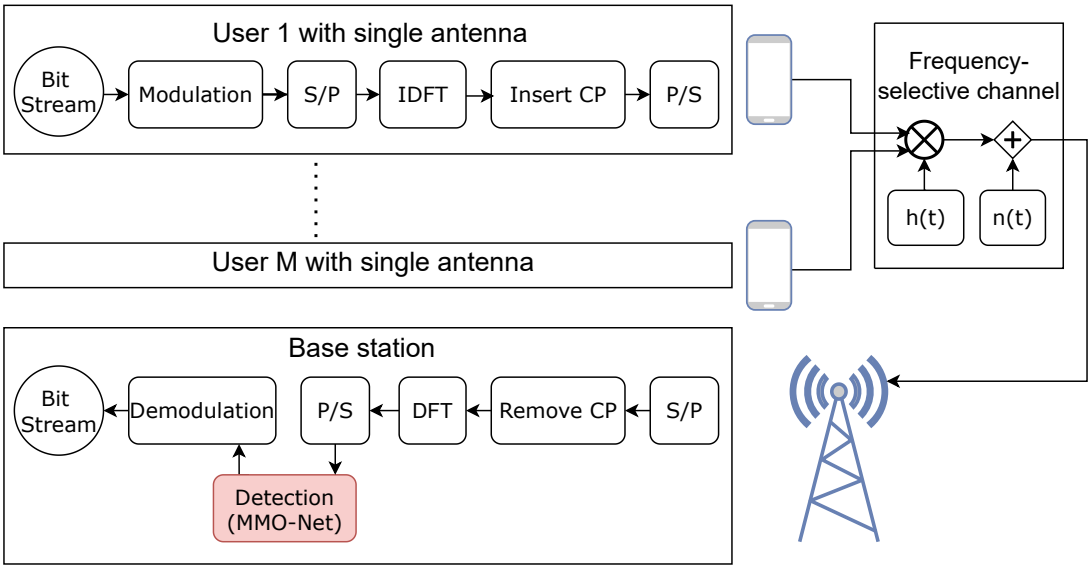


Figure 4.1: The considered MU-MIMO-OFDM uplink system with the proposed detection network.

4.2 System Model

For reliable communication over frequency-selective channels, in this chapter, we propose new detection schemes for a massive MU-MIMO uplink system that combines with OFDM. Here, we consider an uncoded MU-MIMO-OFDM system, where the channel state information (CSI) is unknown to the transmitters. Generally, an MU-MIMO-OFDM communication system includes three elements: M mobile user terminals with a single antenna, multipath channels, and the base station equipped with N antennas. The architecture of our system is shown in Fig. 4.1.

For each user, the original bits are mapped into a sequence of modulated symbols. These symbols are converted into parallel streams and processed by inverse discrete Fourier transform (IDFT). Then, the cyclic prefix (CP) that is not shorter than the maximum delay spread of channels is inserted in front of the time-domain symbols. After adding CP, the M single-antenna user terminals transmit signals simultaneously to a BS. On the BS side, CP removal and the discrete Fourier transform (DFT) are

conducted first to transform the received signals into the frequency domain. The DFT length is W, which is also the number of subcarriers in an OFDM symbol. For each subcarrier $\omega \in \{1, \cdots, W\}$, the transmit-receive relationship can be expressed as:

$$\boldsymbol{y}^{\omega} = \boldsymbol{H}^{\omega}\, \boldsymbol{x}^{\omega} + \boldsymbol{n}^{\omega} \tag{4.1}$$

where $\boldsymbol{H}^{\omega} \in \mathbb{C}^{N \times M}$ is the MIMO channel matrix, $\boldsymbol{n}^{\omega} \in \mathbb{C}^{N}$ is complex white Gaussian noise vector with zero mean and variance N_0, $\boldsymbol{y}^{\omega} \in \mathbb{C}^{N}$ is the vector of received signals, and $\boldsymbol{x}^{\omega} \in \chi^{M}$ is the transmitted symbol vector. The finite alphabet set of constellation points is indicated as χ, where every symbol has a uniform probability of being chosen by the transmitters. In our system, the constellation set χ is provided by QAM modulation schemes.

In order to enable DL-based solutions, we rewrite all the complex-valued vectors and matrices in (4.1) in real-valued versions:

$$\begin{aligned} \boldsymbol{y}^{\omega} &= \begin{bmatrix} \Re\{\boldsymbol{y}^{\omega}\} \\ \Im\{\boldsymbol{y}^{\omega}\} \end{bmatrix}, \quad \boldsymbol{H}^{\omega} = \begin{bmatrix} \Re\{\boldsymbol{H}^{\omega}\} & -\Im\{\boldsymbol{H}^{\omega}\} \\ \Im\{\boldsymbol{H}^{\omega}\} & \Re\{\boldsymbol{H}^{\omega}\} \end{bmatrix}, \\ \boldsymbol{x}^{\omega} &= \begin{bmatrix} \Re\{\boldsymbol{x}^{\omega}\} \\ \Im\{\boldsymbol{x}^{\omega}\} \end{bmatrix}, \quad \boldsymbol{n}^{\omega} = \begin{bmatrix} \Re\{\boldsymbol{n}^{\omega}\} \\ \Im\{\boldsymbol{n}^{\omega}\} \end{bmatrix} \end{aligned} \tag{4.2}$$

where the real-valued $\boldsymbol{y}^{\omega} \in \mathbb{R}^{2N}$, $\boldsymbol{H}^{\omega} \in \mathbb{R}^{2N \times 2M}$, $\boldsymbol{x}^{\omega} \in \mathbb{R}^{2M}$, and $\boldsymbol{n}^{\omega} \in \mathbb{R}^{2N}$. To avoid the complex notation of MU-MIMO-OFDM systems obfuscating the key features of our algorithm architecture, we omit the subcarrier index ω in the following derivation process. In Chapter 4 which focuses on data detection schemes, the channel matrix $\boldsymbol{H}$ is assumed to be perfectly known at the BS. To recover the $\boldsymbol{x}$ from the received

observations $\boldsymbol{y}$, the detection rule of ML is represented by:

$$\widehat{\boldsymbol{x}}_{ML} = \arg\min_{x \in \mathbb{C}^M} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 \tag{4.3}$$

4.3 The ADMM Architecture for MIMO Detection

It is well known that the computational complexity of the ML detection in (4.3) is exponential in M, which is prohibitive in massive MU-MIMO systems. In practice, alternative detectors with different complexity are proposed by researchers [31]. The signal detection problem can be generalized from the ML (4.3) as:

$$\widehat{\boldsymbol{x}} = \mathcal{J}(\boldsymbol{y}, \boldsymbol{H}) \tag{4.4}$$

In this work, we consider the ADMM algorithm which is powerful for large-scale structured optimization problems as the basic skeleton of the proposed neural network. Based on the ADMM architecture, the $\mathcal{J}()$ in (4.4) can be designed specifically for MIMO detection tasks. For this purpose, we first introduce the generic ADMM method in Subsection 4.3.1, and then derive the MIMO detector based on ADMM architecture in Subsection 4.3.2.

4.3.1 The Generic ADMM Method

To better explain the ADMM algorithm, we first describe the equality-constrained convex optimization problem as:

$$\begin{aligned} &\text{minimize} \quad f(\boldsymbol{x}) \\ &\text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = b \end{aligned} \tag{4.5}$$

where $f()$ is a convex function, and $\boldsymbol{x} \in \mathbb{R}^n$ is the variable. For the problem (4.5), the corresponding Lagrangian is:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\mu}^T (\boldsymbol{A}\boldsymbol{x} - b) \tag{4.6}$$

where $\boldsymbol{\mu}$ is a dual variable. As one of the predecessors of ADMM algorithms, the dual ascent method is developed to solve the problem (4.5):

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \underset{\boldsymbol{x}}{\operatorname{argmin}} \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{\mu}^k\right) \\ \boldsymbol{\mu}^{k+1} &= \boldsymbol{\mu}^k + \alpha\left(\boldsymbol{A}\boldsymbol{x}^{k+1} - b\right) \end{aligned} \tag{4.7}$$

where k is the iteration index, and α is the step size. The method of multipliers has very similar iterations to dual ascent except for the $\boldsymbol{x}$-update step based on the augmented Lagrangian. The goal of the Augmented Lagrangian is to improve the robustness and convergence of dual ascent methods, especially without assumptions such as strict convexity.

As described in [92], ADMM is an algorithm that combines the decomposability of the dual ascent method with the superior convergence properties of the method of multipliers. Here we rewrite the optimization problem (4.5) as a regularized estimation problem in an ADMM form:

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = c \end{aligned} \tag{4.8}$$

where $f()$ is the estimation loss, and $g()$ can be interpreted as the regularization term. As an improved version of the method of multipliers, the ADMM is also based on the augmented Lagrangian. The augmented Lagrangian associated with (4.8) is presented

by:

$$\mathcal{L}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\mu}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \boldsymbol{\mu}^T(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - c) + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - c\|_2^2 \tag{4.9}$$

The augmented Lagrangian over the $\boldsymbol{x}$ and $\boldsymbol{z}$ variables can be solved by the standard ADMM iterations:

$$\begin{aligned}
\boldsymbol{x}^{k+1} &= \underset{\boldsymbol{x}}{\operatorname{argmin}} \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{z}^k, \boldsymbol{\mu}^k\right) \\
\boldsymbol{z}^{k+1} &= \underset{\boldsymbol{z}}{\operatorname{argmin}} \mathcal{L}\left(\boldsymbol{x}^{k+1}, \boldsymbol{z}, \boldsymbol{\mu}^k\right) \\
\boldsymbol{\mu}^{k+1} &= \boldsymbol{\mu}^k + \rho\left(\boldsymbol{A}\boldsymbol{x}^{k+1} + \boldsymbol{B}\boldsymbol{z}^{k+1} - c\right)
\end{aligned} \tag{4.10}$$

where ρ is the penalty parameter in the augmented Lagrangian (4.9), which is used as the step size in ADMM iterations. For the sake of convenience, the standard ADMM (4.10) is often written in the scaled form:

$$\begin{aligned}
\boldsymbol{x}^{k+1} &= \arg\min_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}) + \frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z}^k - c + \boldsymbol{u}^k\right\|_2^2 \right\} \\
\boldsymbol{z}^{k+1} &= \arg\min_{\boldsymbol{z}} \left\{ g(\boldsymbol{z}) + \frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x}^{k+1} + \boldsymbol{B}\boldsymbol{z} - c + \boldsymbol{u}^k\right\|_2^2 \right\} \\
\boldsymbol{u}^{k+1} &= \boldsymbol{u}^k + \boldsymbol{A}\boldsymbol{x}^{k+1} + \boldsymbol{B}\boldsymbol{z}^{k+1} - c
\end{aligned} \tag{4.11}$$

where $\boldsymbol{u}^k = \boldsymbol{\mu}^k/\rho$ is the scaled dual variable. Here, the $\boldsymbol{x}$-update involves minimizing $f()$, the $\boldsymbol{z}$-update involves minimizing $g()$, and the last step is the dual variable update. Different from the method of multipliers which minimizes the augmented Lagrangian (4.9) jointly with respect to the two variables, the ADMM algorithm (4.11) updates $\boldsymbol{x}$ and $\boldsymbol{z}$ in an alternating way. That is why it is called alternating direction method of multipliers.

4.3.2 ADMM-Based MIMO Detection

The ADMM architecture we introduced in the last subsection can be used to solve many optimization problems. In this subsection, we will derive a MIMO detection algorithm based on the scaled ADMM iterations. If the estimation loss $f(\boldsymbol{x})$ is considered as the form of (4.3) for MIMO detection tasks, the regularized estimation problem (4.8) can be rewritten as:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|_2^2+g(\boldsymbol{z}) \\ &\text{subject to} \quad \boldsymbol{z}=\boldsymbol{x} \end{aligned} \tag{4.12}$$

The augmented Lagrangian associated with (4.12) is presented by:

$$\mathcal{L}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{u})=\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|_2^2+g(\boldsymbol{z})+\frac{\rho}{2}\|\boldsymbol{x}-\boldsymbol{z}+\boldsymbol{u}\|_2^2 \tag{4.13}$$

To compute the first step of the ADMM-based MIMO detector, the derivative of (4.13) with respect to the variable $\boldsymbol{x}$ is set to 0 as follows:

$$\boldsymbol{H}^H(\boldsymbol{H}\boldsymbol{x}-\boldsymbol{y})+\rho(\boldsymbol{x}-\boldsymbol{z}+\boldsymbol{u})=0 \tag{4.14}$$

Combined with the new $\boldsymbol{x}$-update derived from (4.14), the scaled ADMM iterations (4.11) are redesigned to perform the minimization of the augmented Lagrangian for MIMO detection (4.13) over the variables $\boldsymbol{x}$ and $\boldsymbol{z}$ as follows:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \left(\boldsymbol{H}^H\boldsymbol{H}+\rho\boldsymbol{I}\right)^{-1}\left[\boldsymbol{H}^H\boldsymbol{y}+\rho\left(\boldsymbol{z}_k-\boldsymbol{u}_k\right)\right] \\ \boldsymbol{z}_{k+1} &= \text{proj}\left(\boldsymbol{x}_{k+1}+\boldsymbol{u}_k\right) \\ \boldsymbol{u}_{k+1} &= \boldsymbol{u}_k-\boldsymbol{z}_{k+1}+\boldsymbol{x}_{k+1} \end{aligned} \tag{4.15}$$

where I is an identity matrix that has the same size as $\boldsymbol{H}^H\boldsymbol{H}$, and ρ is a step size, the single parameter of the ADMM algorithm. Note that the $g()$ in (4.13) is the indicator function of the constellation set χ which is a closed nonempty convex set. Thus, the $\boldsymbol{z}$-update of the ADMM-based detector can be solved by a non-linear projection function of $(\boldsymbol{x}_{k+1}+\boldsymbol{u}_k)$ onto χ, i.e. by using proj(), which will be discussed in detail in Subsection 4.4.3.

4.4 MMO-Net: A Deep Unfolding Network for MIMO-OFDM Detection

In the era of 5G and beyond, many new communication applications require high throughput and low latency, which means the affordable number of algorithm iterations or network layers should be relatively small. On the one hand, some prior conventional MIMO detectors or state-of-the-art model-driven networks achieve fast convergence and low complexity. However, with only several fixed or trainable parameters in total, the architectures of these schemes are too inflexible to adapt to the systems with impairments and realistic correlated channels. On the other hand, those pure data-driven solutions have large models with millions of learnable parameters depending on the system size, which makes them hard to be well-trained for massive MIMO-OFDM systems. For data detection tasks, a good trade-off between performance and complexity is one of the foremost goals. Thus, in this section, we propose a DU-based MIMO detector with a moderate number of parameters and flexibility as well as the corresponding model optimization strategies.

4.4.1 Neural Network Architecture Based on OR-ADMM

In the ADMM-based detection scheme (4.15), there is only one algorithm parameter, i.e. the step size ρ. Moreover, the value of ρ is fixed between conventional ADMM iterations. These characteristics lead to the very limited degrees of freedom of the ADMM-based detector. Intuitively, there are two ways that can increase the model flexibility and improve the detection performance: adding proper parameters into the original architecture and optimizing different sets of parameters for different iterations. Based on these two design principles, we first revisit the (4.15) with consideration of the over-relaxation parameter. According to [91], the OR-ADMM is able to obtain a guaranteed improvement in terms of convergence compared to the classical ADMMs. By replacing all the terms $\boldsymbol{x}_{k+1}$ in the $\boldsymbol{z}$- and $\boldsymbol{u}$-updates of the proposed ADMM-based MIMO detector (4.15) with the terms $(\alpha_k \boldsymbol{x}_{k+1} + (1-\alpha_k)\boldsymbol{z}_k)$, the new iterations with the over-relaxed design are given by:

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \left(\boldsymbol{H}^H\boldsymbol{H} + \rho\boldsymbol{I}\right)^{-1}\left[\boldsymbol{H}^H\boldsymbol{y} + \rho\left(\boldsymbol{z}_k - \boldsymbol{u}_k\right)\right] \\
\boldsymbol{z}_{k+1} &= \text{proj}\left(\alpha\boldsymbol{x}_{k+1} + (1-\alpha)\boldsymbol{z}_k + \boldsymbol{u}_k\right) \\
\boldsymbol{u}_{k+1} &= \boldsymbol{u}_k - \boldsymbol{z}_{k+1} + \alpha\boldsymbol{x}_{k+1} + (1-\alpha)\boldsymbol{z}_k
\end{aligned} \tag{4.16}$$

where $\alpha > 1$ is the over-relaxation parameter, which is empirically advantageous compared to the normal relaxation parameter $\alpha \in (0,2)$ [93]. Compared with the iterations in (4.15), the OR-ADMM-based MIMO detector (4.16) is a better skeleton for our neural networks, while its improvement heavily relies on the choice of the step size ρ and over-relaxation parameter α. For the algorithm parameters with fixed values like ρ and α, conventional methods including analytical [91] or manual [94] selections can obtain the optimal values. However, it is very challenging for these traditional approaches to find the best set of algorithm parameters for different iterations and subcarriers in MIMO-OFDM systems. As an alternative, deep

unfolding (DU) unrolls algorithm iterations as the layers of a deep neural network and introduces some learnable parameters. Instead of conducting exhaustive trials to search for optimal values, these parameters can be learned jointly using deep learning techniques (stochastic gradient descent (SGD)-based optimizers, loss functions, and back propagation), which can efficiently overcome the difficulty of high-dimensional parameter selection. These findings motivate us to develop MMO-Net, a DU-based detection network for massive MU-MIMO-OFDM systems.

4.4.2 MMO-Net Design

Based on the algorithm architecture we derive in (4.16), MMO-Net is proposed for MU-MIMO-OFDM detection as shown in Algorithm 1:

Algorithm 1 MMO-Net for MIMO-OFDM Detection

Preprocessing: $\boldsymbol{h} = \boldsymbol{H}^H \boldsymbol{y}; \quad \boldsymbol{A} = \boldsymbol{H}^H \boldsymbol{H} + \sigma \boldsymbol{I}; \quad \boldsymbol{B} = \boldsymbol{A}^{-1};$

Reorganize $\boldsymbol{h}$ and $\boldsymbol{B}$ as the real form using Eq. (4.2)

Input: $\boldsymbol{h}$ and $\boldsymbol{B}$

Initialization: $\boldsymbol{z}_0 = 0; \quad \boldsymbol{u}_0 = 0$

for $\omega = 1 : W$ **do**
  for $k = 0 : (T-1)$ **do**
    $\boldsymbol{x}_{k+1}^{\omega} = \boldsymbol{B}^{\omega} \left(\boldsymbol{h}^{\omega} + \rho_k^{\omega} \left(\mathbf{z}_k^{\omega} - \boldsymbol{u}_k^{\omega} \right) \right)$
    $\mathbf{z}_{k+1}^{\omega} = \text{proj} \left(\alpha_k^{\omega} \boldsymbol{x}_{k+1}^{\omega} + \left(1 - \alpha_k^{\omega} \right) \boldsymbol{z}_k^{\omega} + \boldsymbol{u}_k^{\omega} \right)$
    $\boldsymbol{u}_{k+1}^{\omega} = \boldsymbol{u}_{\boldsymbol{k}}^{\omega} - \delta_k^{\omega} \left(\mathbf{z}_{k+1}^{\omega} - \alpha_k^{\omega} \boldsymbol{x}_{k+1}^{\omega} - \left(1 - \alpha_k^{\omega} \right) \boldsymbol{z}_k^{\omega} \right)$
  end for
end for
Output: $\widehat{\boldsymbol{z}}_T$

In Algorithm 1, k is the layer index, ω is the subcarrier index, T is the number of layers in MMO-Net, and W is the number of subcarriers in an OFDM symbol. Each layer in MMO-Net corresponds to one algorithm iteration. In the matrix A, $\sigma = N_0/E_s$ denotes the reciprocal of the signal-to-noise ratio (SNR). Note that all preprocessing

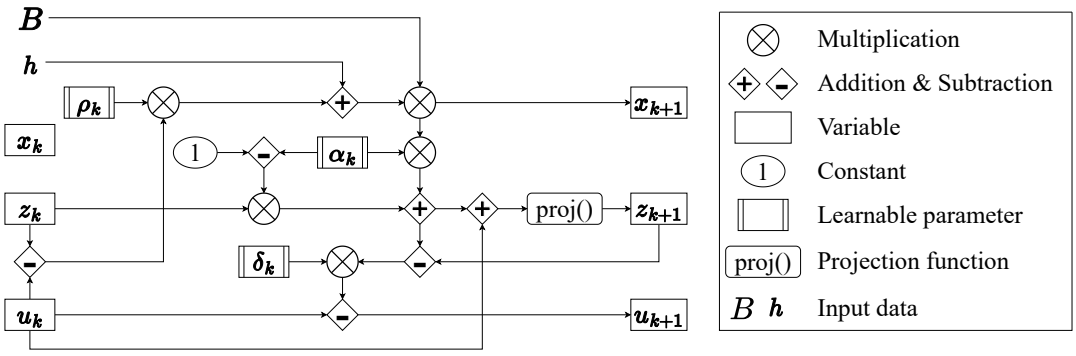


Figure 4.2: Block diagram of the proposed MMO-Net detector.

operations use the complex-valued data, i.e., $\boldsymbol{H}$ and $\boldsymbol{y}$. After preprocessing, $\boldsymbol{h}$ and $\boldsymbol{B}$ are reorganized in the real-valued form following (4.2) and then packaged into training and testing datasets. Fig. 4.2 illustrates the block diagram of the proposed MMO-Net detector after preprocessing.

To obtain further improvement in terms of convergence and performance by utilizing DU techniques, the MMO-Net in Algorithm 1 has many different features from the OR-ADMM-based MIMO detector derived in (4.16). First, to introduce the noise information in the $\boldsymbol{x}$-update, we replace ρ in the matrix $(\boldsymbol{H}^H\boldsymbol{H}+\rho\boldsymbol{I})$ of (4.16) with $\sigma = 1/SNR$, see the definition of $\boldsymbol{A}$ in Algorithm 1. Since the σ is fixed between different layers, the matrix inverse $\boldsymbol{B}$ only needs to be calculated once in the preprocessing stage. In fact, this design is inspired by the widely-used linear MMSE detector. The basic idea of ZF and LMMSE detectors is to calculate the preliminary estimation $\widehat{\boldsymbol{x}}$ of $\boldsymbol{x}$ via linear equalization, i.e., multiplying $\boldsymbol{y}$ by a receive filter $\boldsymbol{G}$. With different $\boldsymbol{G}$, the ZF and LMMSE detection algorithms are given as:

$$\widehat{\boldsymbol{x}}_{ZF} = \boldsymbol{G}_{ZF}\boldsymbol{y} = \left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\boldsymbol{H}^H\boldsymbol{y} \tag{4.17a}$$

$$\widehat{\boldsymbol{x}}_{LMMSE} = \boldsymbol{G}_{LMMSE}\boldsymbol{y} = \left(\boldsymbol{H}^H\boldsymbol{H}+\sigma\boldsymbol{I}\right)^{-1}\boldsymbol{H}^H\boldsymbol{y} \tag{4.17b}$$

Unlike the ZF detector (4.17a), the LMMSE detector (4.17b) takes account of the noise information σ and thus has improved performance. In the first layer of Algorithm 1, i.e., when $\boldsymbol{z}_0 = \boldsymbol{u}_0 = 0$, the $\boldsymbol{x}$-update performs the LMMSE estimation as (4.17b). If σ is unknown, the $\boldsymbol{x}$-update of the first iteration can still perform the ZF detection as initialization.

Second, according to the iterations in (4.16), there is only one tunable step size ρ in the $\boldsymbol{x}$-update. To increase the flexibility to adapt to ill-conditioned channel matrices, we add an additional trainable step size δ into the $\boldsymbol{u}$-update. More precisely, ρ can be interpreted as a regularization parameter in the linear estimation of $\boldsymbol{x}$, and δ is the augmented Lagrangian parameter. Except for the aforementioned ρ, α and δ, there is another learnable parameter λ in the projection function proj(), and it will be described in detail in the next subsection.

Third, different from the conventional OR-ADMM detector (4.16) that has fixed-valued parameters ρ and α, MMO-Net can optimize different sets of learnable parameters for each different iteration k and subcarrier ω. Generally, the iterative algorithms with fixed parameters like the CG [82] and the ADMM (4.15-4.16) are very restrictive. By contrast, some DU-based schemes with layer-wise distinct parameters (e.g. ρ_k) like ADMM-Net [85] have more degrees of freedom. In our massive MU-MIMO-OFDM system, two channel models with different levels of correlation and frequency selectivity are considered, thus we develop two different settings of learnable parameters to provide different levels of model flexibility. For the simple i.i.d. Gaussian channel widely used by prior works [84]-[89], only a single set of 4 learnable parameters $\{\rho_k, \alpha_k, \delta_k, \lambda_k\}_{k=0}^{T-1}$ is adequate for MMO-Net to handle all the channel realizations at different subcarriers. For the challenging 3GPP-3D realistic channel, different sets of parameters, i.e. $\left\{\{\rho_k^\omega, \alpha_k^\omega, \delta_k^\omega, \lambda_k^\omega\}_{k=0}^{T-1}\right\}_{\omega=1}^{W}$, are trained to

adapt to the channel realizations $\{\boldsymbol{H}^{\omega}\}_{\omega=1}^{W}$ at different subcarriers. Compared with the subcarrier-wise identical parameters $\{\rho_k, \alpha_k, \delta_k, \lambda_k\}_{k=0}^{T-1}$, the subcarrier-wise distinct parameters can further improve detection performance due to more expressiveness of the neural network. In this case, the total number of trainable parameters is $4TW$.

A proper set of initial parameters can accelerate the convergence and improve the performance of MMO-Net. Unlike the step size ρ which takes the reciprocal of SNR as the initial value, i.e. $\rho_0^{\omega} = \sigma$, the proper initial values of α and δ are not explicitly relevant to the information of input data. If the over-relaxation parameter $\alpha = 1$, MMO-Net with the OR design will degrade to a classical ADMM-based detection network. According to the analysis in [91], when the step size takes the optimal value, $\alpha = 2$ is the corresponding best option for regularized quadratic minimization problems. Here, we empirically choose $\alpha_0^{\omega} = 1.5$ which is between 1 and 2, and $\delta_0^{\omega} = 2$.

4.4.3 The Differentiable Projection Function

For conventional ADMM-based detection schemes, the projection functions in the iterations are generally not continuously differentiable, such as the sign() used in [94]. To enable the stochastic gradient descent-based optimization used for parameter update, the projection operation proj() in the $\boldsymbol{z}$-update of MMO-Net must be differentiable. Furthermore, one major difference between data-driven DNNs and linear algorithms is the activation function that introduces the nonlinearity leading to better ability of expression. For example, the DNN-based schemes in the literature outperform conventional detection algorithms in different wireless systems, especially with non-linear distortion. Similarly, the non-linear operators can help achieve better performance for model-driven detection networks like DetNet [84]. In [85], ADMM-Net uses the similar soft-sign function as DetNet with an additional learnable offset β as follows:

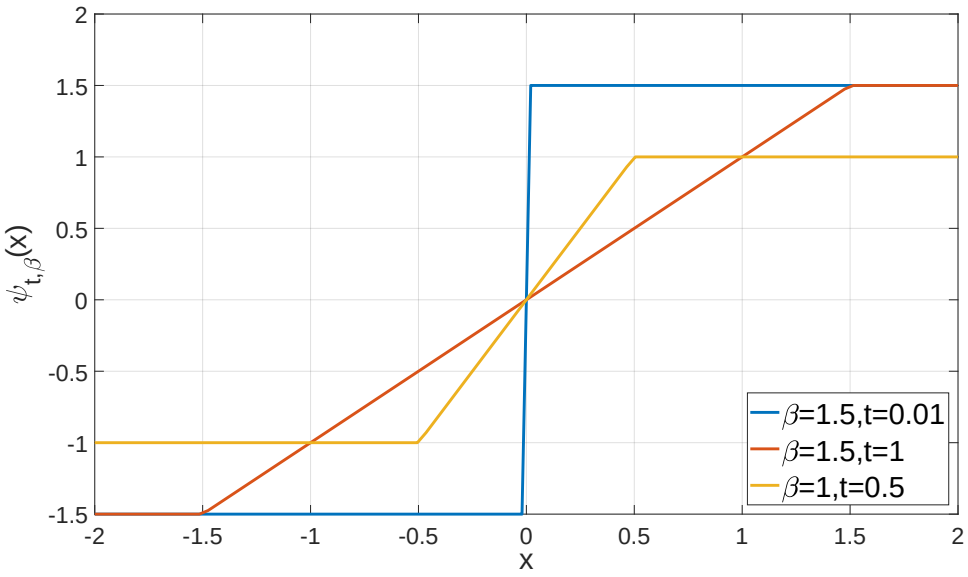


Figure 4.3: The illustration of the piecewise soft-sign operator in ADMM-Net.

$$\psi_{t,\beta}(x) = \frac{\rho(x+\beta t)}{t} - \frac{\rho(x-\beta t)}{t} - \beta \tag{4.18}$$

where $\rho()$ denotes the rectified linear unit (ReLU). The piecewise soft-sign operator (4.18) is depicted in Fig. 4.3. It is obvious that the projection function in (4.18) does not consider the multilevel modulated symbols, which makes them only suitable for low-order modulation schemes like Binary Phase-shift keying (BPSK) and 4-QAM. To solve this limitation, a multilevel activation is proposed in [86]:

$$\sigma_{\mathrm{c}}(x) = \sum_{i=1}^{M} \sigma_{\mathrm{s}}(x-\tau_i) + C \tag{4.19}$$

where τ_i is the sigmoid shift, and C is an overall offset. τ_i and C are fixed and given by the constellation set. By combining several sigmoids $\sigma_s()$ with τ_i and C, the projection (4.19) has multiple levels to adapt to higher-order modulations. Compared with the sigmoid $\sigma_s()$ used in (4.19), the hyperbolic tangent tanh() generally converges faster

due to its bigger gradients. In addition, tanh() with the zero-centered nature is more suitable for high-order (16 or higher) QAM modulation and the received symbols in this chapter. Thus, based on the tanh(), we propose a non-linear projection that does not need the additional fixed or learnable offsets like in (4.18) or (4.19):

$$\begin{aligned} \text{proj}(x;\lambda) &= \sum_{i=1}^{|\chi_r|-1} \tanh\left(\lambda\left(\boldsymbol{x}-\tau_i\right)\right) \\ \tau_i &= \frac{1}{2}\left(s_i+s_{i+1}\right), \quad i=1,\cdots,|\chi_r|-1 \end{aligned} \tag{4.20}$$

where $s_i \in \chi_r$, and $s_i < s_{i+1}$. For the 16-QAM modulation, the value set of both the real and imaginary parts is $\chi_r = \{-3,-1,1,3\}$. In this case, the projection function proj() in (4.20) can be expressed as:

$$\text{proj}(x) = \tanh\left(\lambda\left(\boldsymbol{x}-2\right)\right) + \tanh\left(\lambda\boldsymbol{x}\right) + \tanh\left(\lambda\left(\boldsymbol{x}+2\right)\right) \tag{4.21}$$

where λ is the learnable parameter. With different parameter values, the curves of proj() designed for 16-QAM are illustrated in Fig. 4.4. As shown in the figure, the smoothing coefficient λ controls the smoothness degree of proj(). λ is initialized to 2. After the projection operation, all $\boldsymbol{z}_{k+1}^{\omega}$ in Algorithm 1 are constrained within [-3, 3], which is an ideal range of the final detection output $\widehat{\boldsymbol{z}}_T$. This proper constraint can improve detection performance. Note that the proj() in (4.21) can be easily extended to other modulation schemes by using different combinations of the tanh() with adjustable parameters.

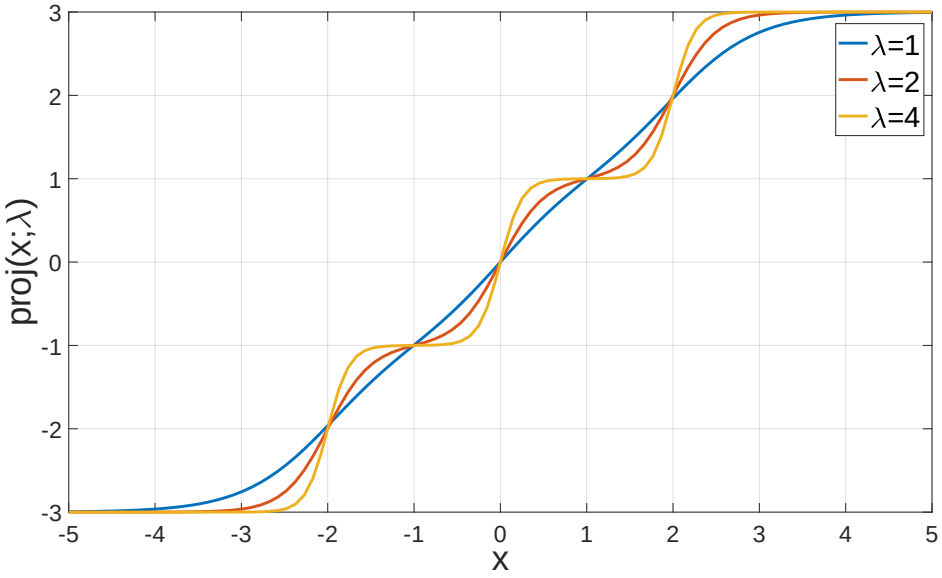


Figure 4.4: Learnable projection proj() with the constraint for 16-QAM.

4.4.4 The Comparison with ADMM-Net

Similar to different detection networks based on the AMP algorithm like [88]-[89], MMO-Net has many differences from ADMM-Net [85], even if they are all derived from the widely used ADMM approach. First, as discussed in the last subsection, the projection function of ADMM-Net (4.18) is designed for low-order modulation schemes. For the massive MIMO-OFDM systems with 16-QAM, MMO-Net significantly outperforms ADMM-Net with the help of the multilevel projection proj() in (4.21), as shown in Section 4.5. Second, the process by which we derive the $\boldsymbol{x}$-update is different from ADMM-Net, as shown in (9) in [85] and (4.14) in Subsection 4.3.2. Thus, in the $\boldsymbol{x}$-update, ADMM-Net introduces the channel power by a penalty vector $\boldsymbol{\lambda}$ which leads to extra complexity, while the proposed MMO-Net only has a scalar ρ as the step size. Moreover, ADMM-Net only has one trainable parameter ω except for the projection function, which limits its flexibility and convergence speed. On the contrary, MMO-Net has not only two step sizes ρ and δ that can increase the flexibility of

ADMM-based architecture but also the over-relaxed parameter α for fast convergence. Furthermore, MMO-Net can learn subcarrier-wise distinct parameters to adapt to those challenging realistic channels with serious correlations and ill-conditioned channel matrices. The simulation results in Fig. 4.5 demonstrate that MMO-Net performs better and needs much fewer iterations to converge than ADMM-Net, especially when the numbers of Tx and Rx antennas are equal.

4.4.5 Computational Complexity

Table 4.1 compares the trainable parameter number and computational complexity of state-of-the-art deep unfolding-based detectors in massive MU-MIMO-OFDM systems. For the DU-based networks with real inputs, the number of real-valued multiplications is defined as computational complexity. The M, N, T, W in Table 4.1 indicate the number of transmitting and receiving antennas, layers, and subcarriers, respectively. For DetNet in [84], the computational complexity of each layer is $O\left(M^2\right)$ due to the matrix-vector products. Due to the large number of learnable parameters depending on the system size (M, N), DetNet needs a long time to be well-trained. In Table I, OAMP-Net2 [88] has the highest computational complexity, dominated by the matrix inverse operations required for each layer. By contrast, not only the proposed MMO-Net has relatively low complexity, but also its parameter number is only determined by the number of layers and subcarriers rather than system sizes. These advantages are especially helpful for large-scale MIMO systems. Concretely, in MMO-Net, the matrix-vector multiplication in the $\boldsymbol{x}$-update is dominant in computational complexity, while the rest scalar-vector products are negligible. Hence, the computational complexity of each layer in MMO-Net is only $O\left(M^2\right)$. As mentioned in the last subsection (Subsection 4.4.4), ADMM-Net [85] has one more vector-vector product in the $\boldsymbol{x}$-update, so it is slightly more complex than the proposed MMO-Net.

Table 4.1: Parameter number and complexity of DU-based detection networks.

Detectors	Trainable parameters	Computational Complexity
DetNet [84]	$(6MN+2N+M)TW$	$O\left(M^2TW\right)$
ADMM-Net [85]	$3TW$	$O\left(M^2TW\right)$
OAMP-Net2 [88]	$4TW$	$O\left(M^3TW\right)$
MMO-Net	$4TW$	$O\left(M^2TW\right)$

4.5 Simulation Results

In this section, we first described the channel models used in this paper and the implementation details of our detection networks. Then, we compare and analyze the performance of MMO-Net with the state-of-the-art DU-based detectors [57], [85], [88] under both the i.i.d. Gaussian channels and the realistic 3GPP-3D channels. Next, some simulation experiments are conducted to demonstrate the robustness of MMO-Net to parameter sharing, as well as high or even full user load. Finally, we develop an efficient implementation scheme to further reduce the complexity of MMO-Net. All the numerical results in this section are obtained without channel coding.

4.5.1 Implementation Details

The performance of both conventional and DL-based detectors highly depends on the type of MIMO channel. As discussed before, most prior DU-based networks such as [84]-[89] demonstrate their performance under the normal or correlated i.i.d. Gaussian channels. However, it is practically more important for MIMO detectors to perform robustly with challenging realistic channels. Thus, we evaluate MMO-Net with 16-QAM modulation for two channel models including the realistic 3GPP-3D channels. For the i.i.d. Gaussian channels, each entry of channel matrix $\boldsymbol{H}$ is independently sampled from a zero-mean Gaussian distribution with variance $(1/N)$.

Table 4.2: Configuration of the realistic 3GPP-3D channel model.

Parameters	Values
Scenario	UMa-NLOS
Center frequency	2.68GHz
Bandwidth	60MHz
Number of subcarriers	512
Height of BS tower	30 m
User range from BS	60° sector for $50-1000$ m
Height and speed of users	1.6 m; 10 km/h

Moreover, for the case of urban cellular deployments with 5G base stations and massive mobile user terminals, we generate realistic channel data from the 3GPP-3D MIMO channel model [95]. Based on an open-source channel simulator QuaDRiGa (version v2.4.0) [29], our channel generator of the 3GPP-3D realistic channels is implemented in Matlab. The detailed parameter settings are listed in Table 4.2. In the table, UMa-NLOS specifies the urban-macrocell scenario and non-line-of-sight transmission. In our massive MU-MIMO-OFDM system, there are 512 subcarriers in an OFDM symbol within 60 MHz bandwidth, and the CP length is 32. Thus, the size of each complex channel H is $(N, M, 512)$. A rectangular array with 32 or 64 antennas is equipped on a BS to serve 32 single-antenna mobile users who are randomly distributed into the sector. Considering both walking and driving users in urban areas, we set the average speed to 10 km/h. The channel noise $\boldsymbol{n}$ is sampled from a zero-mean i.i.d. Gaussian distribution whose variance is related to the system SNR defined as $\text{SNR} = \mathbb{E}\|\boldsymbol{H}\boldsymbol{x}\|_2^2 / \mathbb{E}\|\boldsymbol{n}\|_2^2$.

To evaluate the robustness to different SNRs, we train a single MMO-Net and set the testing SNRs in a wide range from 2 dB to 30 dB. For every training batch, the SNR values of 200 samples are uniformly distributed in [2dB, 30dB] with a 4 dB interval. For a fair comparison, all DL-based detectors in our simulations are trained

by 8000 packets of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{H})$ with the label $\boldsymbol{x}$ and tested by 1000 packets per SNR that are independent of the training dataset. Here, each packet contains an OFDM symbol. Moreover, the analysis and initialization of MMO-Net's trainable parameters are elaborated in Section 4.4. As an adaptive optimizer, Adam [75] only needs an initial learning rate (LR). Our simulations suggest that Adam with exponential decay of LR outperforms that with a fixed initial LR. Thus, we choose an Adam optimizer with exponential decay in this chapter, and the initial LR of the exponential decay is set to 0.0005. Moreover, the summation of L2 loss functions over all T layers is adopted as:

$$l_2 = \frac{1}{S} \sum_{s=1}^{S} \sum_{k=1}^{T} \left\| \boldsymbol{x}_s - \hat{\boldsymbol{z}}_{k,s} \right\|_2^2 \tag{4.22}$$

where s indexes different training samples, k indexes different layers, and S is the total number of training samples. All the operations before MMO-Net detection, including the channel generation, the function blocks in our MU-MIMO-OFDM system (Fig. 4.1), and the data preprocessing in Algorithm 1 are implemented with complex-valued data in Matlab. The proposed MMO-Net as well as the DL-based competitors are all trained and tested in the Tensorflow backend.

4.5.2 Detection Performance under i.i.d. Gaussian Channels

In Chapter 4, the system with M users and N receive antennas is denoted by $M \times N$ MIMO channels, and perfect CSI is assumed. $M = N = 32$ is a fully loaded case that is very challenging for massive MIMO detectors and is rarely considered by the existing studies, especially with realistic channels. Thus, we set the layer number of all the DU-based detectors as $T = 10$ when $M \times N = 32 \times 64$ and increase the T to 20 to handle the high interference under 32×32 MIMO cases. In this subsection, we first compare and analysis the convergence speed of ADMM-Net and MMO-Net. Then the bit error

rate (BER) performance of the following MIMO detectors is compared in the massive MU-MIMO-OFDM system with i.i.d. Gaussian channels:

- MMSE: A widely-used linear detector with matrix pseudo-inverse and noise information.
- CG: An AMI-based MIMO detector that iteratively updates the estimation of signals [82]. Here, the CG has fixed 20 iterations.
- LcgNet: An up-to-date detection network [57] based on the CG algorithm. It has 2 trainable step sizes per layer.
- ADMM-Net: An ADMM-based detection network [85] that outperforms DetNet [84] under BPSK or 4-QAM modulation.
- OAMP-Net2: A state-of-the-art MIMO detector based on OAMP unfolding with a matrix inverse operation in each layer [88]. Based on the Github code [96] of OAMP-Net2, we adapt it to our 16-QAM multiuser MIMO system.
- MMO-Net: Our proposed DU-based network designed for MU-MIMO-OFDM detection tasks, as detailed in Section 4.4.
- SIMO: The BER lower bound that has perfect multiuser detection without interference, which is equivalent to single-input-multiple-output detection.

4.5.2.1 Convergence Property

For both conventional iterative MIMO detectors and layer-wise detection networks, there is a trade-off between the BER performance and time complexity. Basically, As the number of iterations and layers increases, a lower error rate is expected. Fig. 4.5 depicts the BER versus the number of layers of MMO-Net and ADMM-Net under i.i.d. Gaussian channels. For the 32×64 MIMO system operating at 16 dB, MMO-Net converges in less than 10 layers, while ADMM-Net needs more than 20 layers to converge. Moreover, even with only 5 layers, MMO-Net still has lower BER than the converged ADMM-Net with 25 layers. For the fully-loaded case of 32×32 system,

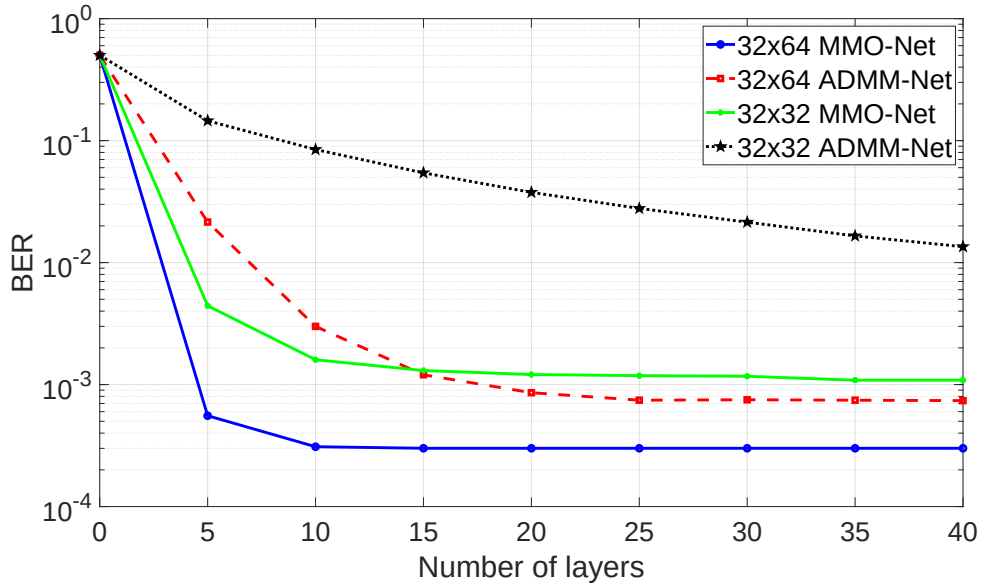
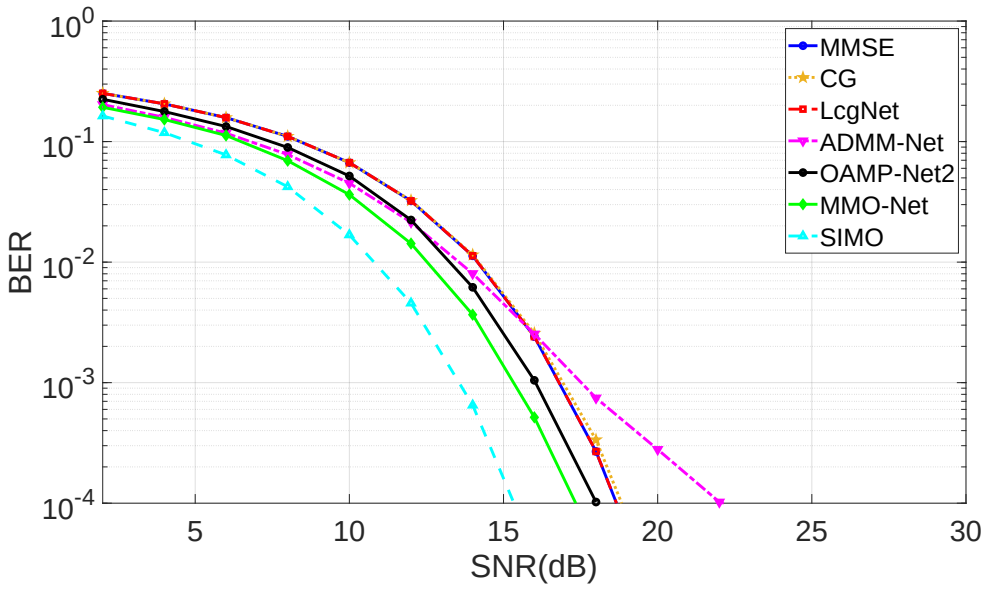


Figure 4.5: BER versus the number of layers of MMO-Net and ADMM-Net.

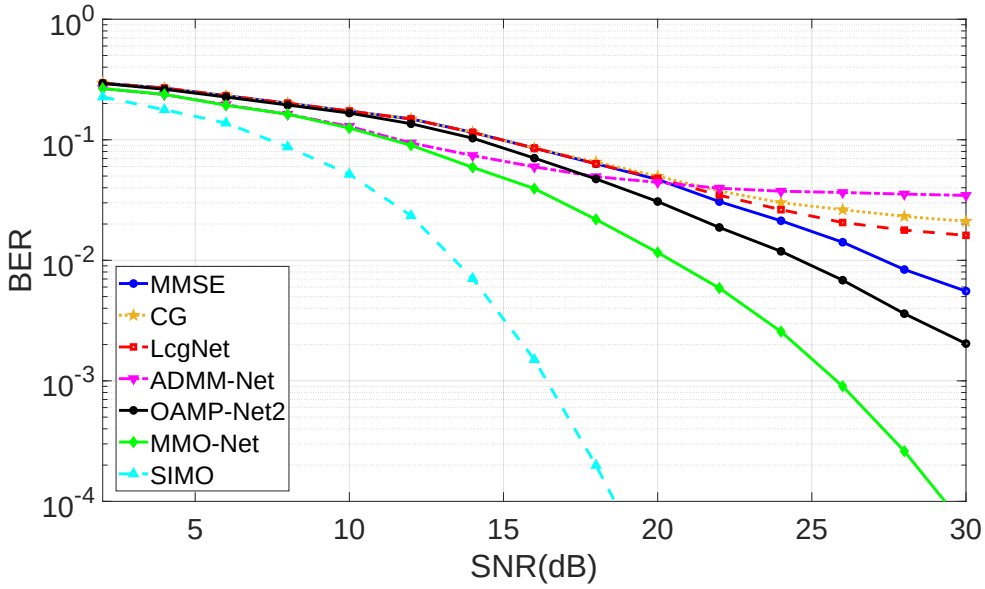
we test the both networks under a higher SNR (22 dB) to get an acceptable BER performance. In this challenging scenario, ADMM-Net cannot converge and achieve an acceptable BER performance (10^{-2}) even with 40 layers. By contrast, MMO-Net almost converges in 15 layers and provides significantly better performance. This is because the projection operator of ADMM-Net (4.20) only works well for low-order modulation up to 4-QAM. MMO-Net has not only the multilevel projection (4.21) with a proper constraint but also the architecture with the OR design, which jointly result in faster convergence and performance gain of MMO-Net under 16-QAM, especially when $M = N$.

4.5.2.2 Detection Performance

In Fig. 4.6, the BER performance of the conventional and four state-of-the-art MIMO detectors based on deep unfolding is compared under i.i.d. Gaussian channels. For all four DL-based detectors, only a common set of trainable parameters for all subcarriers



(a) 32 users, 64 antennas at BS



(b) 32 users, 32 antennas at BS

Figure 4.6: BER versus SNR curves of state-of-the-art MIMO detection schemes for different user loads of the MIMO-OFDM system under i.i.d. Gaussian channels.

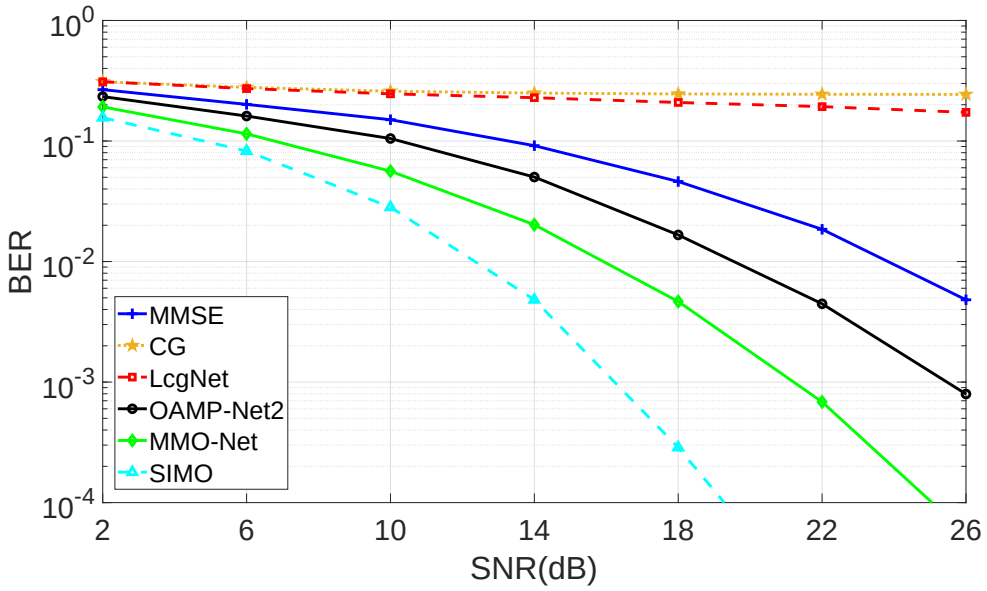
is sufficient to handle this simple channel model. From Fig. 4.6 (a), the CG has near-MMSE performance. With the same architecture, LcgNet outperforms CG, which demonstrates the superiority of deep unfolding. However, the linear LcgNet is still unable to outperform MMSE, which is the performance limit of CG-based detection algorithms. By learning optimal parameters, the OAMP-Net2 with a non-linear denoiser outperforms MMSE for all SNRs. Particularly, the linear estimation step of OAMP-Net2 is able to provide the ideal input for its denoiser under i.i.d. Gaussian channels, which results in competitive performance. When SNR > 16 dB, ADMM-Net performs worst among all schemes since it has no multilevel projection for 16-QAM. Except for the ideal SIMO lower bound, MMO-Net has lower BER than all competitors over all SNRs, including the more complex OAMP-Net2 requiring a matrix inverse per layer.

To test the robustness to the fully loaded case, we evaluate all detection algorithms for 32×32 MIMO channels in Fig. 4.6 (b). First, unlike in the 32×64 MIMO system, here the AMI-based LcgNet cannot achieve near-MMSE performance even with learnable parameters. For the SNRs below 18 dB, ADMM-Net provides competitive performance. Whereas at higher SNRs, OAMP-Net2 outperforms ADMM-Net, LcgNet, and MMSE. Compared to that in Fig. 4.6 (a), the BER performance of ADMM-Net, OAMP-Net2, and MMSE is also obviously degraded in Fig. 4.6 (b), even with the matrix inversion. This is because they suffer a huge performance loss under the serious interference caused by a full load of MIMO systems. In this case, MMO-Net shows impressive performance that is far better than other schemes, especially when SNR > 16 dB.

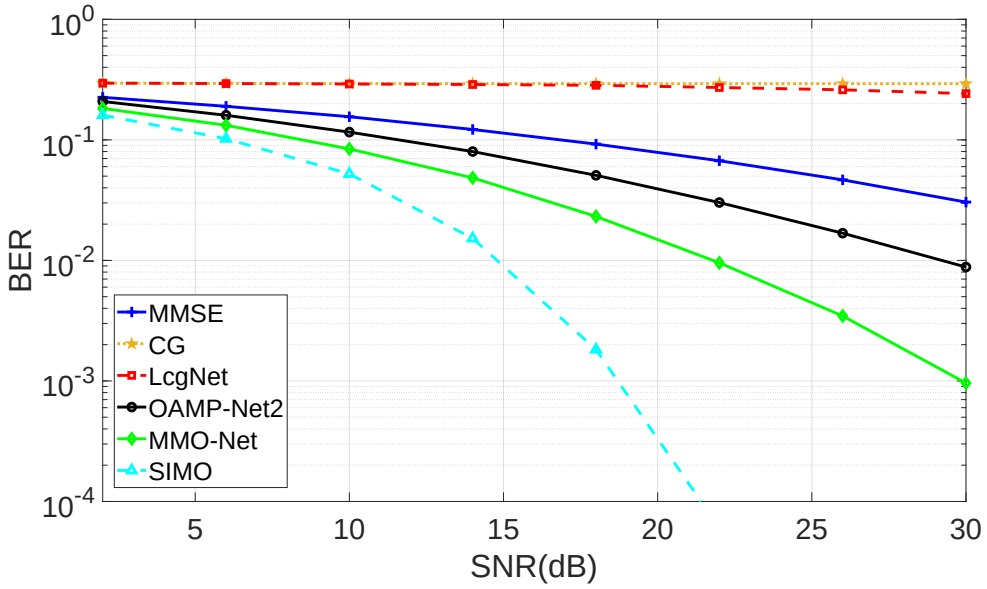
4.5.3 Detection Performance under Realistic 3GPP-3D Channels

Fig. 4.7 depicts the BER performance on realistic 3GPP-3D channels for 32×64 high-loaded and 32×32 full-loaded MIMO systems. Note that we tried to train the ADMM-Net to adapt to our 3GPP-3D channels but met great difficulty in terms of convergence. Thus, in Fig. 4.7, we only consider OAMP-Net2 which outperforms ADMM-Net in Fig. 4.6. For both OAMP-Net2 and MMO-Net, different sets of learnable parameters for each subcarrier are trained to better adapt to this practical channel with serious fading and correlation, as described in Subsection 4.4.2.

In Fig. 4.7 (a), CG fails to detect transmitted symbols under 3GPP-3D channels. LcgNet has a limited performance gain but can no longer provide near-MMSE performance like in Fig. 4.6 (a). Compared to the MMSE with channel matrix inversion, the BER performance of AMI-based algorithms like CG is very unstable and highly depends on the channel models and system settings. Even with the power of deep unfolding, the simple and linear CG architecture of LcgNet limits its generalization ability to achieve better performance under challenging scenarios. Furthermore, thanks to our training strategy with subcarrier-wise learnable parameters, the non-linear OAMP-Net2 obviously outperforms MMSE. In Fig. 4.7 (a), MMO-Net has a much larger performance gap with the more complex OAMP-Net2 than that in Fig. 4.6 (a). For example, when we aim to BER $= 10^{-3}$, MMO-Net only has a 0.7 dB gap with OAMP-Net2 in Fig. 4.6 (a), while this gap increases to 3.5 dB in Fig. 4.7 (a). This is because OAMP-Net2 performs very well with unitarily-invariant channel matrices, but its performance seriously degrades on ill-conditioned realistic channels that do not satisfy this assumption. In contrast, the proposed MMO-Net does not rely on strict assumptions on channel matrices, which leads to its better robustness to realistic channel models.



(a) 32 users, 64 antennas at BS



(b) 32 users, 32 antennas at BS

Figure 4.7: BER versus SNR of state-of-the-art MIMO detection schemes for different user loads of the MIMO-OFDM system under realistic 3GPP-3D channels.

In Fig. 4.7 (b), the performance gap between the SIMO lower bound and all other detectors increases for the 32×32 MIMO system compared to the 32×64 MIMO case in Fig. 4.7 (a). The reason is that they suffer a huge performance loss under the serious interference caused by the full user load. Specifically, the BER of the powerful OAMP-Net2 is only $= 10^{-2}$ under 29 dB SNR, and MMSE can not even achieve an acceptable BER (10^{-2}) within 30 dB. However, in this very challenging case, MMO-Net shows state-of-the-art performance that is much better than other schemes. For instance, MMO-Net has 3-6.5 dB performance gain over the second-best OAMP-Net2 with higher complexity. In summary, with the help of the special-designed projection function, over-relaxed design and well-chosen learnable parameters, the superior performance and robustness of MMO-Net are demonstrated under full load scenarios with realistic channel models.

4.5.4 Efficient Implementation in the Frequency Domain

In the era of 5G and beyond, MIMO-OFDM has become a promising scheme for high spectral efficiency wideband systems. Nevertheless, MIMO-OFDM receivers are computationally demanding since the signal processing is performed on a tone-by-tone (subcarrier) basis. Thus, an interpolation-based efficient matrix inversion method is proposed for MIMO-OFDM receivers in [97]. However, the complexity reduction of this scheme becomes not very significant when the number of receive antennas is bigger than 5, which makes it unsuitable for our massive MIMO-OFDM system. Moreover, in modern MIMO-OFDM systems, the number of subcarriers per symbol is generally much larger than 64 in [12]. Thus, we increase the subcarrier number to 512 in this chapter. As mentioned above, learning different sets of parameters for each subcarrier can help MMO-Net perform better under realistic channels with correlation and ill-conditioned channel matrices. Whereas as the increase of subcarriers, the number of trainable parameters and training overhead will increase correspondingly.

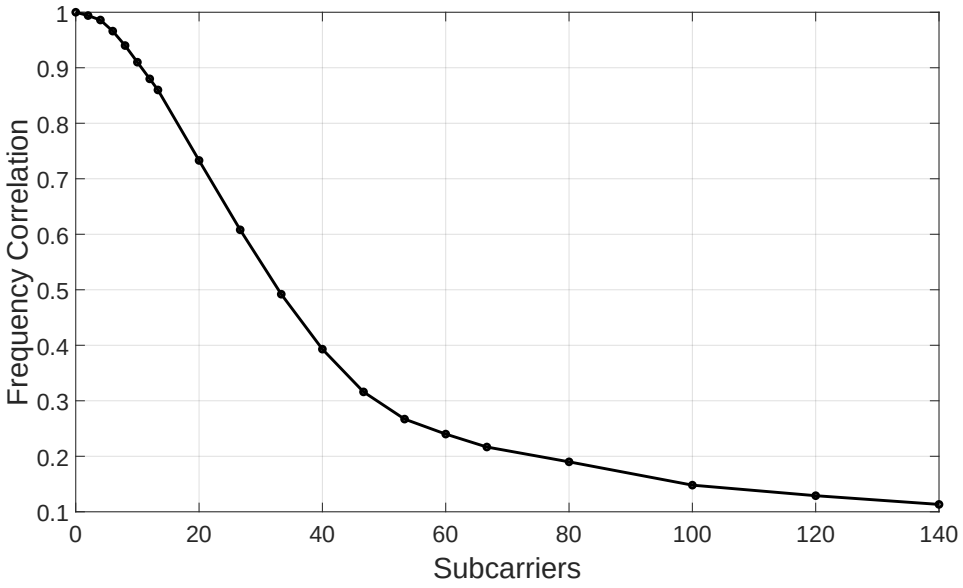


Figure 4.8: Frequency correlation of 3GPP channel realizations over subcarriers.

In order to reduce the training overhead and computational complexity, we further develop an efficient implementation scheme by exploiting both the generalization ability of MMO-Net and frequency correlation of channels. In this subsection, we first explore the correlation of 3GPP-3D channel realizations in the frequency domain. To numerically represent the frequency correlation between channel matrices at different subcarriers, we compute the inner product of different $\boldsymbol{H}^{\omega}$ and then utilize the matrix norm to normalize it. As a result, the frequency correlation between $\boldsymbol{H}^{\omega}$ and itself equals 1. Within 16 adjacent subcarriers, the frequency correlation of channel matrices is very strong (> 0.8), and it then decreases rapidly, as shown in Fig. 4.8.

Based on the results of channel correlations in the frequency domain in Fig. 4.8, it is reasonable to expect that MMO-Net is able to generalize its superior performance from a trained channel matrix at one subcarrier to those untrained channel matrices within a range of subcarriers. Therefore, we introduce the idea of parameter sharing between different subcarriers into the implementation of MMO-Net. Specifically, MMO-Net-n

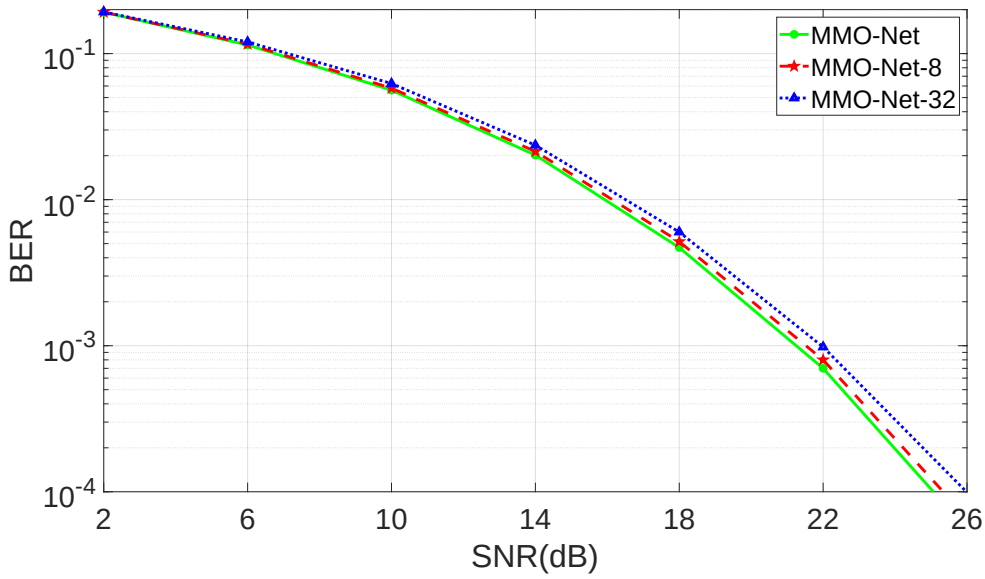


Figure 4.9: BER performance comparisons of MMO-Net with different levels of parameter sharing for a 32×64 MIMO system under 3GPP-3D channels.

is trained by the channel realizations at the middle subcarrier of a group with n subcarriers and tested on the channels of the whole subcarrier group.

Fig. 4.9 compares the BER performance of several MMO -Net-n with different levels of parameter sharing for a 32×64 MIMO-OFDM system with 16-QAM modulation. Compared to the MMO-Net without parameter sharing, there is almost no performance loss for the MMO-Net-8 which shares learnable parameters within a small group of subcarriers with strongly correlated channels. As in Fig. 4.8, the frequency correlation of the 3GPP-3D channels decreases drastically from 0.94 at 8 subcarriers to 0.5 at 32 subcarriers. Since MMO-Net-32 shares parameters for a big group including some distant subcarriers, it indeed has a performance loss. However, this performance loss is not huge and acceptable, as shown in Fig. 4.9. Different from those pure data-driven DNNs with overly general architectures and massive trainable parameters, the model-driven MMO-Net has interpretable architecture and guaranteed performance. Hence, MMO-Net can still provide a robust detection performance even without optimal

parameters for each channel realization. In summary, with the help of the efficient implementation scheme in the frequency domain, MMO-Net is able to achieve a good trade-off between performance and complexity for different scenarios and channel models.

4.6 Summary

In this chapter, to ensure reliable communication on realistic channels with spatial correlation and frequency selectivity, we discuss the development of novel DL-based detection schemes for the case when massive MU-MIMO is combined with OFDM. For data detection tasks of MU-MIMO-OFDM systems, a deep unfolding-based neural network, MMO-Net is proposed. This solution combines the powerful ADMM optimization algorithm, expert knowledge of MIMO detection, as well as state-of-the-art deep learning techniques. First, based on the generic ADMM architecture, an iterative MIMO detection algorithm is derived as a skeleton of our model-driven neural network. Next, to obtain further improvement in terms of convergence and performance by utilizing DU techniques, a series of novel designs are developed to increase the degree of freedom of MMO-Net's architecture, such as over relaxation, extra trainable parameters, and two training strategies for different channel models. Then, a differentiable projection function with multiple levels is proposed to enable SGD-based parameter optimization.

To provide more insights into DL-based MU-MIMO-OFDM detection tasks, we implement three state-of-the-art DU-based schemes [57], [85], [88] in our system. Extensive simulations are carried out to compare them with the proposed MMO-Net in terms of complexity, convergence, and performance. According to the simulation results, with similar or lower complexity, MMO-Net outperforms traditional and neural network-based detectors in massive MU-MIMO-OFDM systems. In particular, MMO-

Net shows significant performance gain for the MU-MIMO-OFDM systems with full user load and real-world 3GPP-3D channels. Moreover, this network can be trained to converge much faster than ADMM-Net which is also based on ADMM architectures. Finally, the efficient implementation scheme proposed in Subsection 4.5.4 can further reduce the computational complexity and training overhead of MMO-Net.

Although some challenging cases like fully loaded MIMO-OFDM systems and realistic channels with severe fading and correlations have already been studied in this chapter, the DU-based detectors including MMO-Net assume perfect CSI, which is unrealistic in practice. An efficient channel estimator is essential for the receivers of massive MU-MIMO-OFDM systems. Therefore, the focus now shifts to the development of efficient channel estimation schemes and data detection with channel estimation errors, which will be discussed in detail in the next chapter.

Chapter 5

CNN-Based Channel Estimation for Massive MIMO-OFDM Systems

5.1 Introduction

The explosion of advanced wireless applications, such as intelligent terminal access, virtual reality, augmented reality, and Internet of things, has propelled the development of wireless communication into the fifth generation (5G) [2]. The emerging 5G wireless communication system raises new requirements on spectral efficiency and energy efficiency. As one of the breakthrough technologies of 5G, massive multiple-input multiple-output (MIMO) is a fundamental approach to exploiting spatial domain resources. MIMO offers diversity gain, multiplexing gain, and power gain [19] to improve reliability, support the spatial multiplexing of single and multiple users, and increase energy efficiency [3]. However, all of these improvements rely on sufficient channel knowledge. Thus, channel estimation becomes a bottleneck in system implementation [98].

5.1.1 Literature Review

Generally, traditional channel estimation (CE) methods include least squares (LS) [59], minimum mean-square error (MMSE), and their optimized versions based on different interpolation schemes [60], [99]. For MIMO systems, several channel estimation schemes have been proposed in [100], [101], and [102]. In [100], the authors propose a

data-aided channel estimation method where partially decoded data is used to estimate the channel. Based on compressed sensing, a sparse channel estimation solution is developed in [101]. For frequency division duplex-based massive MIMO systems, a spatially common sparsity-based adaptive channel estimation and feedback scheme is proposed in [102].

MIMO channels are associated with high-dimensional optimization parameters, and the optimization problem itself can be difficult to be solved by conventional solutions. Deep learning (DL)-based technologies offer new possibilities to model the complicated channels of massive MIMO. Due to its data-driven nature, the DL-based system can automatically learn high-level features from raw data instead of manual feature extraction. Recently, DL has been applied to channel estimation in the physical layer of wireless communication systems [103]. In [48], a deep neural network (DNN)-based channel estimator is proposed in an online fashion to adapt to doubly selective (i.e., time and frequency selective) channels. The work of [104] views the channel frequency response (CFR) at the pilot positions as a low-resolution image, and the complete channel information is recovered by two convolutional neural networks (CNN). For high-speed scenarios, a long short-term memory (LSTM) network is cascaded with a CNN to estimate fast-fading channels [105]. Note that these three works are developed for single-input single-output (SISO) orthogonal frequency-division multiplexing (OFDM) systems. For single-input multiple-output (SIMO) systems, preliminary theoretical analysis is presented to interpret the internal mechanisms of DL-based channel estimation [106].

To estimate high-dimensional channel matrices in MIMO systems, the authors replace the one-dimensional (1D) CNN in [105] with a 2D-CNN in [107]. In [108], a spatial-frequency CNN (SF-CNN) based approach is proposed to improve the estimation performance in mmWave MIMO systems. Furthermore, a joint channel estimation and signal detection scheme is designed for point-to-point MIMO systems [88]. In [109], a

single DNN is used for the channel estimation of a single-user MIMO system. For the multiuser (MU) MIMO system in [110], different DNNs are used to estimate multiple individual channels from multiple users separately. However, the schemes in [109] and [110] only consider frequency-flat channels rather than the multi-carrier systems with frequency-selective channels. For multi-carrier MIMO systems like MIMO-OFDM, each tone at each receiver antenna is associated with multiple channel parameters, which makes channel estimation difficult.

5.1.2 Contributions

In Chapter 4, several conventional MIMO detectors, state-of-the-art deep unfolding (DU)-based detection networks, and the proposed MMO-Net are evaluated with perfect channel state information (CSI), which is unrealistic in practice. For proper detection of transmitted signals, CSI should be obtained first via channel estimation. Thus, an efficient channel estimator is essential for the receivers of massive MU-MIMO-OFDM systems. For low-loaded MU-MIMO-OFDM systems like in [111], time-orthogonal pilots are used for all M users. Such pilot schemes lead to low spectral efficiency when M becomes larger. Instead, we design an optimal frequency-orthogonal pilot scheme that only requires one pilot symbol for the 32 users in our massive MU-MIMO-OFDM systems. Furthermore, this chapter proposes three CNN-based channel estimators, i.e., a denoising CNN (D-CNN), a spatial-frequency CNN with a residual layer (RSF-CNN), and an end-to-end DRSF-CNN. The major contributions of this chapter can be summarized as follows:

1. To mitigate channel estimation errors in the delay domain, we propose a compact D-CNN to denoise the channel impulse response (CIR). Instead of learning the mapping to labeled channel matrices directly like usual CNN-based schemes, D-CNN learns the denoising mapping via a special subtractive residual layer to remove the channel noise from the coarse estimate. Compared to the CNN-based

estimator with a widely-used additive residual layer, D-CNN has better mean-square error (MSE) performance.

2. With the aim of improving the estimation performance of D-CNN, we propose RSF-CNN to exploit channel correlation in the spatial-frequency domain. To minimize the overall MSE, we develop a fast Fourier transform (FFT) layer with zero padding to enable end-to-end training of D-CNN and RSF-CNN across different domains. As a result, the proposed DRSF-CNN performs better and converges faster than the state-of-the-art SF-CNN [108].

3. We analyze the computational complexity and the number of trainable parameters of DRSF-CNN in detail, and also provide a layer-wise comparison with SF-CNN to explain why the DRSF-CNN with lower complexity can outperform SF-CNN. Moreover, from an overall perspective of receiver design, we simulate the performance of DRSF-CNN and MMO-Net jointly. To the best of our knowledge, this is one of the first works that jointly evaluate DL-based channel estimation and model-driven detection networks for massive MU-MIMO-OFDM systems with realistic channels.

The remainder of this chapter is organized as follows. In Section 5.2, we first introduce the system model and discuss the prior works on DL-based channel estimation. Next, Section 5.3 develops the frequency-orthogonal pilot scheme and proposes a compact D-CNN to denoise the estimated CIR in the delay domain. To enhance the estimation performance, in Section 5.4, an FFT layer and RSF-CNN are added after the D-CNN to refine the channel frequency response (CFR) in the spatial-frequency domain further. Then, Section 5.5 presents the simulation results to demonstrate the superior convergence and performance of the proposed CNN-based estimators. Finally, Section 5.6 concludes this chapter.

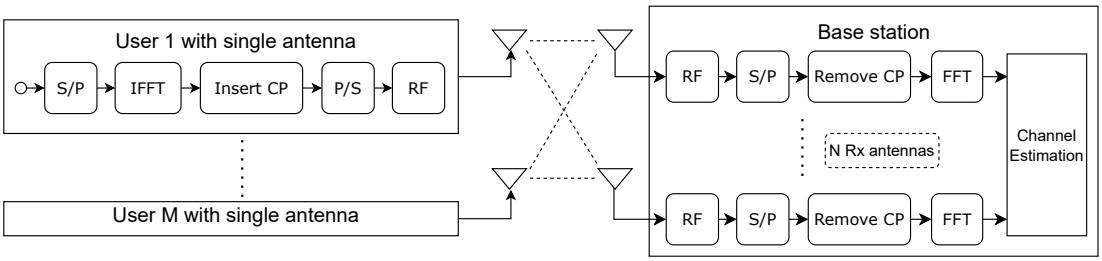


Figure 5.1: Block diagram of a multiuser MIMO-OFDM system with channel estimation.

5.2 Background

In this section, the MU-MIMO-OFDM system for channel estimation is first introduced. Then, we present the existing DNN- and CNN-based channel estimation schemes and discuss the challenges to implementing them in our massive MU-MIMO-OFDM system, including complexity and pilot overhead.

5.2.1 System Model

As shown in Fig. 5.1, we consider a massive MU-MIMO-OFDM system with M single-antenna users and the base station (BS) equipped with N antennas. Suppose the number of subcarriers in each OFDM symbol is W, and the maximum length of all channels is L. For the mth mobile user, the original pilot symbols are processed by inverse fast Fourier transform (IFFT), and the cyclic prefix (CP) that is not shorter than L is added. After removing the CP at the nth receive antenna, the received pilot symbol $\widetilde{\boldsymbol{y}}_p^n \in \mathbb{C}^W$ in the time domain is given by:

$$\widetilde{\boldsymbol{y}}_p^n = \sum_{m=1}^{M} \boldsymbol{H}_{cir}^{n,m} \widetilde{\boldsymbol{x}}_p^m + \widetilde{\boldsymbol{n}}_p^n \tag{5.1}$$

where $\boldsymbol{H}_{cir}^{n,m}$ is a circulant matrix with the first column given by $\left[\boldsymbol{h}^{n,m^T}, 0_{1\times(W-L)}\right]^T$, $\boldsymbol{h}^{n,m} \in \mathbb{C}^L$ is the length L CIR between the mth user and the nth receive antenna, and $\widetilde{\boldsymbol{x}}_p^m \in \mathbb{C}^W$ is the time-domain pilot transmitted by the mth user. Let $\mathcal{F}$ denotes the $W \times W$ normalized FFT matrix, and the eigenvalue decomposition of $\boldsymbol{H}_{cir}^{n,m}$ is:

$$\boldsymbol{H}_{cir}^{n,m} = \mathcal{F}^H \operatorname{diag}\left\{\sqrt{W}\mathcal{F}\left[\boldsymbol{h}^{n,m^T}, 0_{1\times(W-L)}\right]^T\right\}\mathcal{F} \tag{5.2}$$

Based on (5.1) and (5.2), the frequency-domain received symbol $\boldsymbol{y}_p^n$ can be obtained by the FFT of $\widetilde{\boldsymbol{y}}_p^n$ as:

$$\begin{aligned}\boldsymbol{y}_p^n &= \sum_{m=1}^{M} \mathcal{F}\boldsymbol{H}_{cir}^{n,m}\widetilde{\boldsymbol{x}}_p^m + \mathcal{F}\widetilde{\boldsymbol{n}}_p^n \\ &= \sum_{m=1}^{M} \operatorname{diag}\left\{\sqrt{W}\mathcal{F}\left[\boldsymbol{h}^{n,m^T}, 0_{1\times(W-L)}\right]^T\right\}\mathcal{F}\widetilde{\boldsymbol{x}}_p^m + \mathcal{F}\widetilde{\boldsymbol{n}}_p^n \\ &= \sum_{m=1}^{M} \operatorname{diag}\left\{\boldsymbol{F}\boldsymbol{h}^{n,m}\right\}\boldsymbol{x}_p^m + \boldsymbol{n}_p^n\end{aligned} \tag{5.3}$$

where $\boldsymbol{F} \in \mathbb{C}^{W\times L}$ is $\sqrt{W}$ times the first L columns of $\mathcal{F}$. The vectors $\boldsymbol{x}_p^m$ and $\boldsymbol{n}_p^n$ are the pilot signals and noise samples at the FFT output, separately.

5.2.2 Learning-based Channel Estimation Schemes

Recently, DL-based CE solutions have shown the potential to provide a competitive estimation performance without prior channel knowledge. For the SISO-OFDM system in [48], the SIMO system in [106] or single-carrier MIMO systems [109]-[110], the fully-connected DNN-based channel estimators in the literature have good performance and acceptable complexity. However, when the number of antennas and subcarriers becomes large, these DNN-based schemes are hard to realize. Specifically,

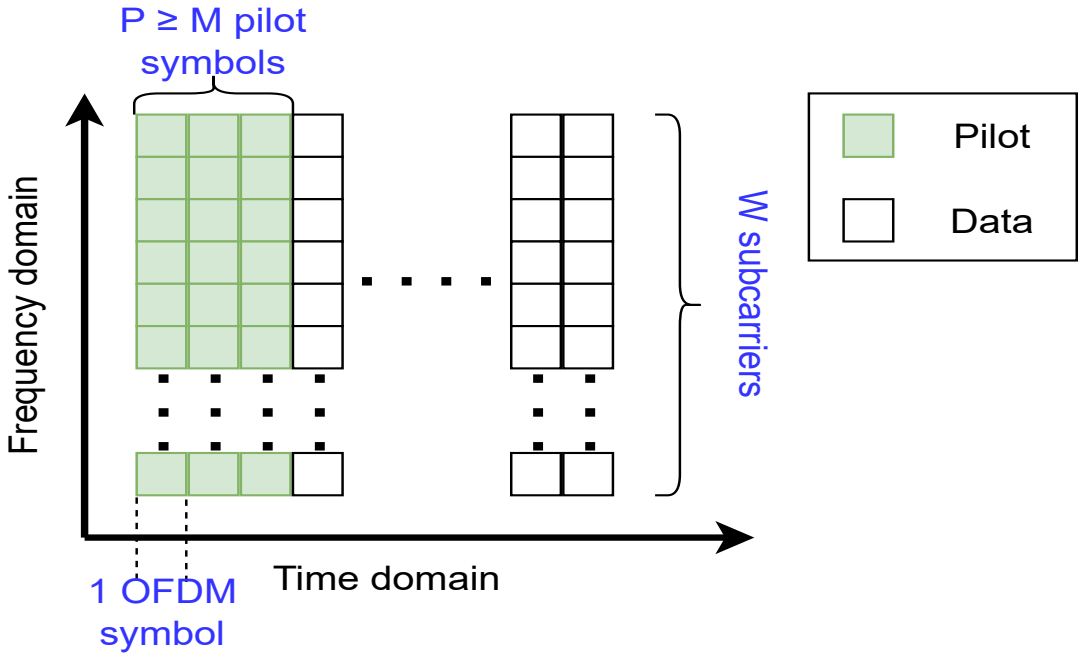


Figure 5.2: The data structure of time-orthogonal pilots transmitted by each single-antenna user in an MU-MIMO-OFDM system with M users.

each fully-connected layer has millions of trainable parameters for the very high dimensional channel matrices in our massive MIMO-OFDM system. Due to the cost of much time and computational resource, those DNNs with such large numbers of trainable parameters are very difficult to train.

By exploiting parameter sharing and sparse interactions, CNN-based schemes show a better performance-complexity trade-off for CE tasks in [104]-[105] and [107]-[108]. Compared to the point-to-point (single-user) MIMO systems in [88], [107] and [108], pilot schemes and channel estimation for MU-MIMO systems are different. Most prior solutions allocate pilots to all users orthogonally over the time domain (i.e., the number of pilot symbols P is larger than the user number M), as shown in Fig. 5.2. Such pilot schemes are capable of low-loaded MU-MIMO-OFDM systems like in [111] with only 8 users, but result in a considerable reduction of spectral efficiency in our system with 32 users. Thus, it is desirable to develop an efficient pilot pattern and corresponding channel estimator for massive high-loaded MU-MIMO-OFDM systems.

5.3 Channel Estimation and Denoising in the Delay Domain

5.3.1 Initial Channel Estimation based on LS

Generally, the LMMSE estimator performs well by utilizing prior channel statistics but is sensitive to imperfect channel information. On the contrary, with the powerful learning ability, DL-based estimators do not restrict to any specific channel statistics. Due to the difficulty of capturing prior information in practice, we choose the LS estimator without prior requirements on channel statistics as the initialization of our proposed channel estimation networks.

In order to derive the LS estimation used in this chapter, we first simplify the transmit-receive relationship in (5.3). By defining $\boldsymbol{A} = \left[\text{diag}\left\{\boldsymbol{x}_p^1\right\}\boldsymbol{F}, \cdots, \text{diag}\left\{\boldsymbol{x}_p^m\right\}\boldsymbol{F}\right] \in \mathbb{C}^{W\times ML}$, equation (5.3) can be rewritten as:

$$\begin{aligned}\boldsymbol{y}_p^n &= \sum_{m=1}^{M} \text{diag}\left\{\boldsymbol{x}_p^m\right\}\boldsymbol{F}\boldsymbol{h}^{n,m} + \boldsymbol{n}_p^n \\ &= \boldsymbol{A}\boldsymbol{H}^n + \boldsymbol{n}_p^n\end{aligned} \tag{5.4}$$

where $\boldsymbol{H}^n = \left[\boldsymbol{h}^{n,1^T}, \cdots, \boldsymbol{h}^{n,M^T}\right]^T$. Thus, we can now derive the LS channel estimate for our MIMO-OFDM system as follows:

$$\widehat{\boldsymbol{H}}_{LS}^n = \boldsymbol{A}^\dagger \boldsymbol{y}_p^n \tag{5.5}$$

where $\boldsymbol{A}^\dagger$ denotes the pseudo-inverse of $\boldsymbol{A}$.

5.3.2 An optimal Pilot Scheme for Massive MU-MIMO with High User Load

In this chapter, we consider an uplink scenario where all mobile users simultaneously transmit their pilots to the BS. As mentioned above, time-orthogonal pilot schemes in the literature will consume substantial time resources in massive MU-MIMO-OFDM systems with high user load. Thus, the pilots that are orthogonal for all users over the frequency domain are considered in our system.

For different system settings, the constraints on optimal pilot sequences are different. For the OFDM systems with a single transmit antenna, the optimal pilot scheme should be equi-powered and equi-spaced [112]. In [113], optimal training signal design for frequency-selective channel estimation in MIMO-OFDM systems is analyzed based on minimizing MSE. For the MIMO-OFDM systems with flat-fading channels ($L = 1$), the optimal pilot sequences on different transmit antennas must be orthogonal. When the channels are frequency selective ($L > 1$), the optimal pilots on different users should be orthogonal not only to the pilot sequences of other users but also to the phase shifts of these sequences. In other words, the optimal pilot sequences for channel estimation based on one or multiple OFDM symbols are shown to be equi-powered, equi-spaced, and phase-shift orthogonal. Furthermore, the authors in [114] present more general optimal training signals for MIMO-OFDM channel estimation.

With unit modulus and good periodic correlation properties, the constant amplitude zero autocorrelation (CAZAC) sequence is able to satisfy the conditions of the optimal pilot design. As a well-known CAZAC sequence, the Zadoff-Chu (ZC) sequence [115] is widely used in wireless communication systems. Based on the ZC sequence, the pilot of the mth user is proposed as:

$$\boldsymbol{x}_p^m(\omega) = \exp\left(-j\frac{\pi u \omega\,(\omega + W \bmod 2 + 2q_m)}{W}\right) \tag{5.6}$$

where $\omega \in \{1, \cdots, W\}$ is the subcarrier index in the pilot symbol, q_m denotes the phase shift of the mth user, and u is an integer that is prime to W. Except for constant amplitude, the special autocorrelation property of ZC-based pilot sequences is particularly useful. Concretely, the autocorrelation of a ZC sequence with a cyclically shifted version of itself is zero, i.e., different cyclically shifted versions of a ZC sequence are orthogonal to one another. Since circular shift orthogonality in the time domain is equivalent to phase shift orthogonality in the frequency domain, the proposed ZC-based pilots in (5.6) meet the conditions of optimal pilot sequences mentioned above.

To obtain the exact form of (5.6), we need to know the proper value of phase shift q_m. To minimize the MSE, the phase shifts of any two users must satisfy the specific constraint like the (22) in [113]. For the scenario with M users and frequency-selective channels with L paths, the phase shift q_m can be set to $(m-1)L$, $m \in \{1, \cdots, M\}$. Let $u = 1$ and $W \mod 2 = 0$, then the simplified version of (5.6) can be written as:

$$\boldsymbol{x}_p^m(\omega) = \exp\left(-j\frac{\pi\omega(\omega + 2(m-1)L)}{W}\right) \tag{5.7}$$

Note that the optimal pilot sequence in [113] is not based on the ZC sequences, which is different from our proposed pilot (5.7). However, this pilot scheme and (5.7) share many similar properties, since it can be viewed as a part of (5.7). We can rewrite (5.7) as:

$$\begin{aligned} \boldsymbol{x}_p^m(\omega) &= \exp\left(-j\frac{\pi\omega^2}{W}\right)\exp\left(-j\frac{2\pi\omega(m-1)L}{W}\right) \\ &= \exp\left(-j\frac{\pi\omega^2}{W}\right)\boldsymbol{b}_p^m(\omega) \end{aligned} \tag{5.8}$$

As mentioned in [113], when $\boldsymbol{b}_p$ is optimal, the product of $\boldsymbol{b}_p$ and an arbitrary sequence with unit modulus like the $\exp\left(-j\frac{\pi\omega^2}{W}\right)$ in (5.8) is also optimal. This conclusion can demonstrate that the proposed pilot sequences in (5.7) and (5.8) are optimal from another perspective.

5.3.3 D-CNN: An Efficient Channel Denoising Network

To improve the performance of channel estimation in massive MU-MIMO-OFDM systems, we first analyze the channel estimation error. For the transmit-receive relationship in (5.4) and the corresponding LS estimation in (5.5), the CE error model can be denoted as:

$$\widehat{\boldsymbol{H}}^n = \boldsymbol{H}^n + \boldsymbol{E}^n \tag{5.9}$$

where $\boldsymbol{E}^n$ indicates the channel estimation error. For most of the existing DL-based channel estimation schemes, the coarse estimate $\widehat{\boldsymbol{H}}^n$ in (5.9) is refined by learning the following mapping function:

$$\mathcal{F}\left(\widehat{\boldsymbol{H}}^n\right) \approx \boldsymbol{H}^n = \widehat{\boldsymbol{H}}^n - \boldsymbol{E}^n \tag{5.10}$$

where the refined estimate $\mathcal{F}\left(\widehat{\boldsymbol{H}}^n\right)$ and $\mathcal{F}()$ are the output and the mapping function of channel estimation networks, separately. For LS channel estimation (5.5) of the CIR, the additive Gaussian noise $\boldsymbol{n}_p^n$ is the dominant source of CE error $\boldsymbol{E}^n$. Thus, proper denoising can significantly improve the estimation performance. Inspired by an efficient image denoising scheme [116], we propose a denoising CNN, D-CNN, to eliminate the CE error $\boldsymbol{E}^n$ in the delay domain. Instead of learning $\mathcal{F}\left(\widehat{\boldsymbol{H}}^n\right)$ directly to predict the accurate channel $\boldsymbol{H}^n$, D-CNN learns the denoising mapping $\mathcal{H}\left(\widehat{\boldsymbol{H}}^n\right) \approx \boldsymbol{E}^n$

to remove $\boldsymbol{E}^n$ from the coarse estimate $\widehat{\boldsymbol{H}}^n$ as follows:

$$\boldsymbol{H}^n = \widehat{\boldsymbol{H}}^n - \mathcal{H}\left(\widehat{\boldsymbol{H}}^n\right) \tag{5.11}$$

To solve equation (5.11) explicitly via D-CNN, a special residual layer with subtraction which takes the original input $\widehat{\boldsymbol{H}}^n$ and the output of the last convolutional layer $\mathcal{H}\left(\widehat{\boldsymbol{H}}^n\right)$ is added at the end of D-CNN. In the original paper on residual learning [62], the residual mapping $\mathcal{R}(x) = \mathcal{F}(x) + x$ is learned via additive residual layers to ease the training of very deep networks. For the CE task in (5.10), the $\mathcal{R}\left(\widehat{\boldsymbol{H}}^n\right)$ is still learned to predict $\boldsymbol{H}^n$ directly, which is different from the $\mathcal{H}\left(\widehat{\boldsymbol{H}}^n\right)$ for explicit denoising. To validate the effectiveness of the specialized subtractive residual layer in D-CNN for channel denoising, the MSE performance of D-CNN and a CNN with a widely-used additive residual layer will be compared in Subsection 5.5.

For the mth user, $\widehat{\boldsymbol{H}}^m_{LS} = \left[\widehat{\boldsymbol{h}}^{1,m^T}_{LS}, \cdots, \widehat{\boldsymbol{h}}^{N,m^T}_{LS}\right]^T \in \mathbb{C}^{NL}$ is the CIR estimated by the LS (5.5). Replacing the $\widehat{\boldsymbol{H}}^n$ in (5.11) by the new input $\widehat{\boldsymbol{H}}^m_{LS}$, the D-CNN for MU-MIMO-OFDM channel estimation can be formulated as:

$$\widehat{\boldsymbol{H}}^m_{DCNN} = \widehat{\boldsymbol{H}}^m_{LS} - \mathcal{H}\left(\widehat{\boldsymbol{H}}^m_{LS}\right) \tag{5.12}$$

where $\widehat{\boldsymbol{H}}^m_{DCNN} \in \mathbb{C}^{NL}$ is the output of D-CNN. In modern OFDM systems, the number of subcarriers W in each symbol is generally much larger than the number of channel taps L in the frequency-selective communications channel. Therefore, the size of $\widehat{\boldsymbol{H}}^m_{LS}$ in the delay domain is much smaller than the channel matrix $\widehat{\boldsymbol{H}}^m_{FFT} \in \mathbb{C}^{NW}$ in the frequency domain. The small-sized input $\widehat{\boldsymbol{H}}^m_{LS}$ leads to the low complexity of compact D-CNN. Note that the complex-valued $\widehat{\boldsymbol{H}}^m_{LS}$ needs to be separated into the real and imaginary parts first, so the input size of D-CNN is $(N, L, 2)$. $(N, L, 2)$ is also the output size of the D-CNN.

For the hidden layers of D-CNN, the size of convolutional filters is set to 3×3, and the rectified linear unit (ReLU) is adopted as the activation function. To get a better trade-off between performance and complexity, choosing a proper number of filters and layers for D-CNN is essential. Generally, increasing the number of filters can improve CNN's performance since the network can learn additional features. Based on our trials, the performance improvement by further increasing the number of convolutional filters from 16 is minor. Thus, each hidden layer of D-CNN has 16 filters. The choice of depth for D-CNN will be illustrated in Section 5.5. Moreover, the last convolutional layer of D-CNN only uses 2 filters, and then its output is subtracted from the original input $\widehat{\boldsymbol{H}}_{LS}^{m}$ in the residual layer, as shown in Fig. 5.3.

5.4 DRSF-CNN: End-to-end Learning across Different Domains

Before elaborating on the different modules of the proposed DRSF-CNN, we first introduce the overall framework. As illustrated in Fig. 5.3, DRSF-CNN consists of a D-CNN proposed in Section 5.3, a customized FFT layer, and an RSF-CNN. Note that the matrix sizes in Fig. 5.3, e.g. $(N, L, 2)$, indicate the output sizes of different layers. With the help of the customized FFT layer, D-CNN and RSF-CNN working in different domains can be trained end to end rather than one by one. Moreover, in this section, the proposed RSF-CNN and DRSF-CNN are introduced in detail. Finally, we compare the complexity of our DRSF-CNN and SF-CNN [108].

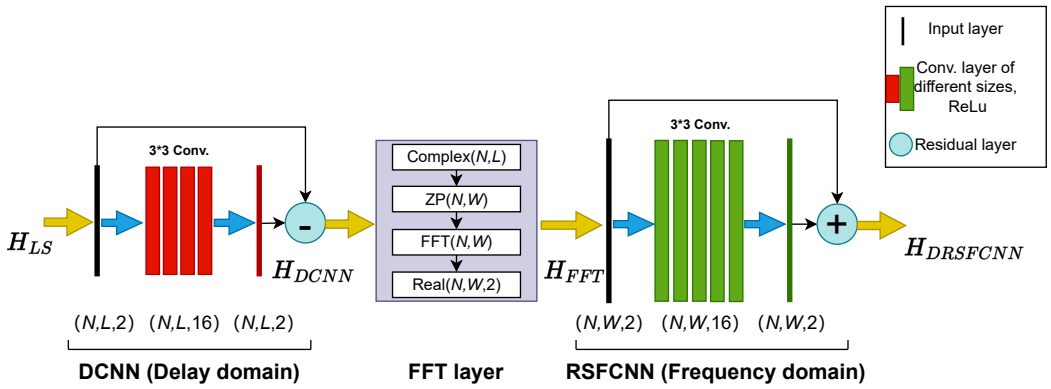


Figure 5.3: The architecture of DRSF-CNN consisting of a D-CNN, a customized FFT layer, and an RSF-CNN.

5.4.1 Network Architecture

Unlike the D-CNN which learns the special mapping $\mathcal{H}()$ to denoise CIR in the delay domain, most existing CNN-based channel estimators like [108] learn the mapping $\mathcal{F}()$ to refine the estimates of CFR in the frequency domain. With the aim of improving estimation performance, the schemes combining several CNNs or LSTMs are proposed in [104], [105], and [107]. However, the data among different layers or networks is always the CFR, which means these solutions can only extract channel features in the frequency domain. To learn the underlying features of channels in both the delay and the frequency domains, RSF-CNN is proposed to further enhance the performance of D-CNN:

$$\mathcal{R}\left(\left(\widehat{\boldsymbol{H}}^{m}_{DCNN}\right)_{\text{zp}}\mathcal{D}\right)=\mathcal{F}\left(\left(\widehat{\boldsymbol{H}}^{m}_{DCNN}\right)_{\text{zp}}\mathcal{D}\right)+\left(\widehat{\boldsymbol{H}}^{m}_{DCNN}\right)_{\text{zp}}\mathcal{D} \tag{5.13}$$

where $\mathcal{D}$ is an FFT matrix, and $\mathcal{R}()$ is the typical residual mapping with an additive residual layer, as in Subsection 5.3.3. Since the output of D-CNN, $\widehat{\boldsymbol{H}}^{m}_{DCNN}\in\mathbb{C}^{NL}$ has a smaller size than the input of RSF-CNN, zero padding is conducted first, i.e., $\left(\widehat{\boldsymbol{H}}^{m}_{DCNN}\right)_{\text{zp}}\in\mathbb{C}^{NW}$. After FFT operation, $\left(\widehat{\boldsymbol{H}}^{m}_{DCNN}\right)_{\text{zp}}$ is converted to $\widehat{\boldsymbol{H}}^{m}_{FFT}=\left(\widehat{\boldsymbol{H}}^{m}_{DCNN}\right)_{\text{zp}}\mathcal{D}$, i.e. the input of RSF-CNN in the frequency domain. The input size

of the RSF-CNN is $(N, W, 2)$.

Similar to the D-CNN, each hidden layer of RSF-CNN has 16 filters of size 3×3 followed by a ReLU activation, and the output layer has 2 filters and no non-linear activation. With a residual layer, a skip connection adding the input to the output is realized in RSF-CNN. The final output size of the RSF-CNN is $(N, W, 2)$.

Different from the multiple cascaded CNNs or LSTMs with the same input and output size in the existing schemes [104] and [107], the output of D-CNN and the input of RSF-CNN have different sizes and are in different domains. To train the D-CNN and RSF-CNN jointly to minimize the overall MSE, a series of operations including zero padding and the FFT needs to be integrated into a layer between these two CNNs. Motivated by this, we develop a customized FFT layer, as shown in Fig. 5.3. In the FFT layer, the output of D-CNN, $\widehat{\boldsymbol{H}}^m_{DCNN}$ is first converted to a complex matrix. After zero padding, the FFT of length W is conducted for each row of the complex channel matrix, which is then reorganized into a real matrix of size $(N, W, 2)$. The end-to-end CE network consisting of a D-CNN, an FFT layer, and an RSF-CNN, is called denoising residual SF-CNN (DRSF-CNN). Combining the D-CNN (5.12) and RSF-CNN (5.13), DRSF-CNN can be expressed as:

$$\widehat{\boldsymbol{H}}^m_{DRSFCNN} = \mathcal{R}\left(\left(\widehat{\boldsymbol{H}}^m_{LS} - \mathcal{H}\left(\widehat{\boldsymbol{H}}^m_{LS}\right)\right)_{\text{zp}} \mathcal{D}\right) \tag{5.14}$$

where $\widehat{\boldsymbol{H}}^m_{DRSFCNN} \in \mathbb{C}^{NW}$ is the output of DRSF-CNN. Note that the input size of DRSF-CNN is $(N, L, 2)$, and the output size is $(N, W, 2)$. The depth of DRSF-CNN will be illustrated in Section 5.5. The MSE loss function used to train the entire DRSF-CNN is:

$$\text{MSE loss} = \frac{1}{S}\sum_{s=1}^{S}\left\|\boldsymbol{H}^m_s - \widehat{\boldsymbol{H}}^m_{DRSFCNN,s}\right\|^2 \tag{5.15}$$

where s indexes the sth sample in a mini-batch and S is the batch size.

5.4.2 Complexity Analysis

In [108], an SF-CNN is proposed to refine the coarse estimation by exploiting channel correlations in the frequency domain. Note that the design and target systems of the SF-CNN and the proposed schemes are totally different. First, the SF-CNN takes the input of size $(N, M, 4)$ to jointly estimate the channels of M transmit antennas in a point-to-point MIMO system. Moreover, SF-CNN only considers the correlation of adjacent 2 subcarriers, and it has 64 filters in each of 9 hidden layers with no residual design. To compare SF-CNN fairly with our schemes in terms of complexity and performance, we adapt its input size to $(N, W, 2)$ for the massive MU-MIMO-OFDM system in this paper.

The comparison results of complexity are shown in Table 5.1, where the number of BS antennas is $N = 64$, the number of channel paths is $L = 20$, and the number of subcarriers is $W = 512$. For the convolutional layers, the number of trainable parameters N_p and the complexity in floating point operations (FLOPs) F_c are calculated by the following functions:

$$N_p = C_i K^2 C_o + C_o \tag{5.16}$$

$$F_c = C_i K^2 C_o N W \tag{5.17}$$

where C_i and C_o are the numbers of input and output channels (i.e., the third dimension of data), respectively. K is kernel size. Note that the layers with index 7, 9 and 16 in DRSF-CNN have no trainable parameters, and their computational complexity can be ignored. The complexity of the FFT layer is only $NW log W$. Moreover, DRSF-CNN has an extra output layer (layer 6), which is much less complex than normal convolutional layers. Therefore, DRSF-CNN can be viewed as an end-to-end network with the same depth (nine hidden layers) as the SF-CNN. According to the numerical results in Table

Table 5.1: The comparison of the number of trainable parameters and computational complexity for the SF-CNN and the proposed DRSF-CNN.

CE schemes	Layer index	Type	Output dimensions	Number of tr. param.	Complexity in FLOPs
SF-CNN [108]	1	Input	(64,512,2)	/	/
	2	1st Conv.	(64,512,64)	1216	37.75M
	$3 \sim 10$	Conv.	(64,512,64)	36928 each	1.21G each
	11	Output	(64,512,2)	1154	37.75M
	Sum			297794	9.76G
Proposed DRSF-CNN	1	Input 1	(64,20,2)	/	/
	2	1st Conv.1	(64,20,16)	304	0.37M
	$3 \sim 5$	Conv. 1	(64,20,16)	2320 each	2.95M each
	6	Output 1	(64,20,2)	290	0.37M
	7	Sub. Res.	(64,20,2)	/	/
	8	FFT	(64,512,2)	/	88.78K
	9	Input 2	(64,512,2)	/	/
	10	1st Conv.2	(64,512,16)	304	9.44M
	$11 \sim 14$	Conv. 2	(64,512,16)	2320 each	75.50M each
	15	Output 2	(64,512,2)	290	9.44M
	16	Add Res.	(64,512,2)	/	/
	Sum			17428	330.56M
one-layer NN	/			4.29G	8.59G

5.1, the number of learnable parameters and FLOPs of the whole DRSF-CNN are only 17428 and 330.56M, which are even less than those of one convolutional layer in SF-CNN. The first reason why DRSF-CNN has much fewer trainable parameters and lower complexity is that it only needs 16 filters per hidden layer to get a competitive CE performance via effective denoising in both the delay and SF domains. Second, the feature map size of the layers $2 \sim 6$ in DRSF-CNN is much smaller than that of the corresponding layers in SF-CNN, as $L = 20 \ll W = 512$.

In Table 5.1, we also calculate the computational complexity and the number of learnable parameters of a channel estimation neural network with only one fully-connected layer. First, 4.29G parameters are significantly more than the parameters of the two CNNs, which means the training of the one-layer NN will cost much more time and resources. Furthermore, the one-layer DNN has comparable complexity with the entire SF-CNN and is much more complex than the proposed DRSF-CNN. Thus, as discussed in Subsection 5.2.2, it is very difficult to implement DNN-based CE schemes in the literature in our massive MIMO-OFDM system.

5.5 Simulation Results

In this section, we first describe the configuration of the channel model and system used in this chapter and introduce the implementation details of our proposed three CE networks. Then, the convergence and MSE performance of the proposed CE networks and the state-of-the-art SF-CNN [108] are compared to demonstrate the superiority of our channel estimation schemes.

5.5.1 Implementation Details

A real-world 3D channel model from the 3rd Generation Partnership Project (3GPP) TR 38.901 [95] is used to generate the channel realizations. Similar to Chapter 4, this channel model is implemented based on an open-source channel simulation tool, QuaDRiGa [29]. Here we adopt the same channel parameters as Table 4.2 so that the DRSF-CNN channel estimator and the MMO-Net detector can be evaluated jointly as an entire receiver. In the high-loaded massive MU-MIMO-OFDM system, there are $M = 32$ single-antenna users that are randomly dropped into the cell, and they are served by a BS equipped with $N = 64$ antennas. The number of channel paths is $L = 20$, and the number of subcarriers is $W = 512$.

As shown in Fig 5.1, all mobile users simultaneously transmit frequency-orthogonal pilots with a CP of length 32 to the BS. The optimal pilot sequence is generated according to equation (5.7) and only occupies one OFDM symbol. All the operations in Fig 5.1 before network training, including the channel generation, the process of pilot transmission, and the LS coarse estimation are implemented with complex-valued data in Matlab. Then, the coarsely estimated channels are reorganized in the real-valued form and packaged into the training and independent testing datasets.

For the proposed three CNNs and the competitor, SF-CNN [108], the training and validation datasets contain 7200 and 800 channel realizations, respectively. Different from [108] where different CNNs are trained separately for different SNRs, we only choose a moderate value of 15 dB as the training SNR and test the trained CNNs with 1000 samples per SNR. The number of training epochs is 400, and the batch size is 40. Moreover, we choose Adam [75] as the optimizer. To accelerate the offline training, we set a decay of learning rate (LR) for the Adam optimizer. The initial LR of the decay is set to 0.001, and this value is halved per 100 epochs.

5.5.2 Impact of Depth and Residual Layers

For a better trade-off between performance and complexity, one essential issue in structure design is to set a proper depth for DRSF-CNN. As shown in Fig. 5.3, there are two sub-networks, D-CNN and RSF-CNN in DRSF-CNN, and they have different residual layers which are also relevant to the choice of network depth. In this subsection, we investigate the impacts of the depth and residual layers on network performance to determine the best architecture of DRSF-CNN and validate its superiority. In our simulations, the MSE performance of the following DL-based channel estimators is compared in the massive MU-MIMO-OFDM system with 3GPP-3D channels:

- D-CNN: The proposed D-CNN that denoises estimated CIR in the delay domain with a subtractive residual layer, as in Subsection 5.3.3. For a fair comparison with other CNNs that output CFR, the FFT layer is integrated into D-CNN.
- D-CNN+: The D-CNN with an additive residual layer.
- SF-CNN: A state-of-the-art CE network with 9 hidden layers [108]. To adapt it to our MU-MIMO-OFDM system, we change its input, remove the tanh activation in the output layer, and maintain other settings.
- RSF-CNN: The proposed CNN-based estimator with an additive residual layer that refines CFR in the SF domain. To show the improvements of the residual layer, it has the same 9 hidden layers and network parameters as SF-CNN.
- DRSF-CNN: Our proposed end-to-end network designed for MU-MIMO-OFDM channel estimation. It combines D-CNN, a customized FFT layer and RSF-CNN, as shown in Fig. 5.3.

First, the impact of network depth for different D-CNNs with or without the residual layer is illustrated in Fig. 5.4. Note that the results are obtained under 15 dB, and the two D-CNNs share the same training dataset and network parameters except for the residual layer. In general, the receptive field of D-CNN is increased as network depth increases, which can help D-CNN make use of the context information in a larger region of the input channel matrices. However, without the residual layer, the MSE of D-CNN meets a floor when its depth is larger than 3, which means the increase in network depth or receptive field does not provide performance gain in this case. By contrast, with the help of the subtractive residual layer, the MSE performance of D-CNN is improved with more hidden layers. Due to the more efficient denoising, the D-CNN with the residual layer significantly outperforms the one without the residual layer. In addition, empirically, as the network depth increases, the efficiency of performance improvement will reduce. From Fig. 5.4, we can see that the degree of

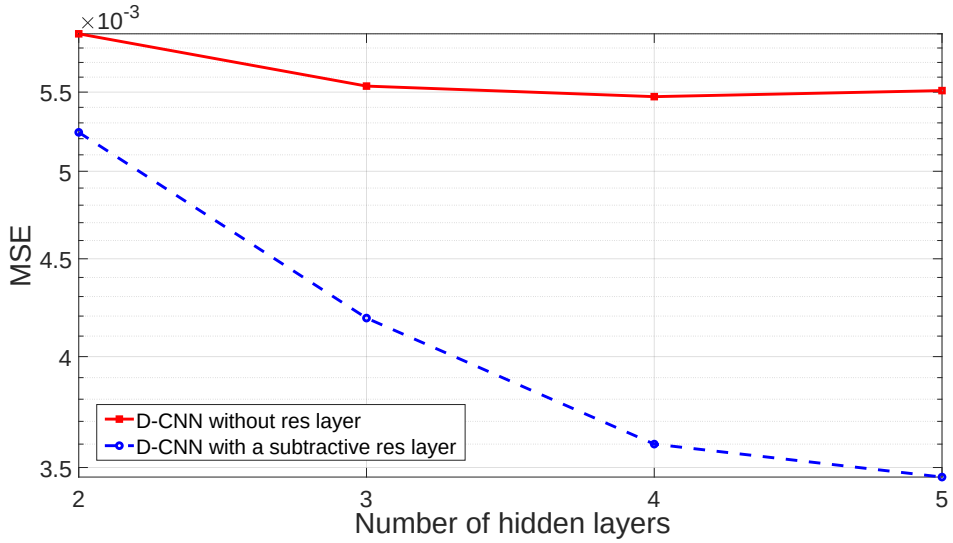


Figure 5.4: The impact of the number of hidden layers for different D-CNNs.

MSE performance improvement for the D-CNN with the residual layer is the greatest when its depth is 4 and then becomes minor when the depth further increases. For a better performance-complexity trade-off, the number of hidden layers in D-CNN is set to 4.

To validate the effectiveness of the denoising mapping $\mathcal{H}()$ in (5.12), in Fig. 5.5, we compare the MSE performance versus training epochs of the baseline D-CNN+ and the proposed D-CNN. In Subsection 5.3.3, we elaborate on the denoising mapping $\mathcal{H}\left(\widehat{\boldsymbol{H}}\right)$ learned by D-CNN and explain the difference between $\mathcal{H}()$ and the widely-used residual mapping $\mathcal{R}()$ in [62]. To compare the estimation performance of these two mappings, we train the D-CNN+ with an additive residual layer to learn the $\mathcal{R}\left(\widehat{\boldsymbol{H}}\right)$. Note that D-CNN+ also has the same 4 hidden convolutional layers as D-CNN. As shown in Fig. 5.5, D-CNN outperforms D-CNN+ in the whole training process. This numerical result demonstrates that the mapping $\mathcal{H}()$ learned via the subtractive residual layer can better denoise the estimated CIR in the delay domain.

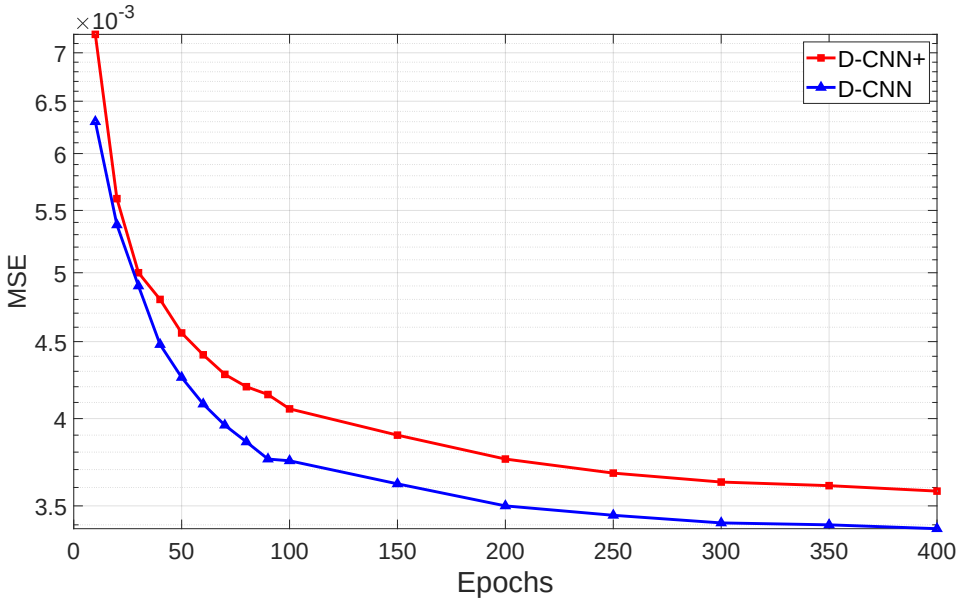


Figure 5.5: Training process of the D-CNN+ and the proposed D-CNN.

According to the results in Fig. 5.4-5.5, the best architecture of the first sub-network of DRSF-CNN is the D-CNN with four hidden layers and a subtractive residual layer. To better understand the effect of the second residual layer and decide the total depth of the DRSF-CNN, we present the MSE performance versus the number of hidden layers of the DRSF-CNN's second sub-network (i.e., RSF-CNN) in Fig. 5.6. In this figure, the DRSF-CNN with the second residual layer always performs better. Without this residual layer, the performance of DRSF-CNN is degraded with the increase in depth. In contrast, with the aid of the second residual layer, DRSF-CNN can obtain accuracy gains from increased depth. According to our experimental trials, further increasing the number of hidden and residual layers as in [62] can not provide significant MSE performance gains for the CE tasks in our massive MU-MIMO-OFDM system. Thus, we choose to use 5 hidden layers in the RSF-CNN. The final architecture of the entire DRSF-CNN is illustrated in Fig. 5.3.

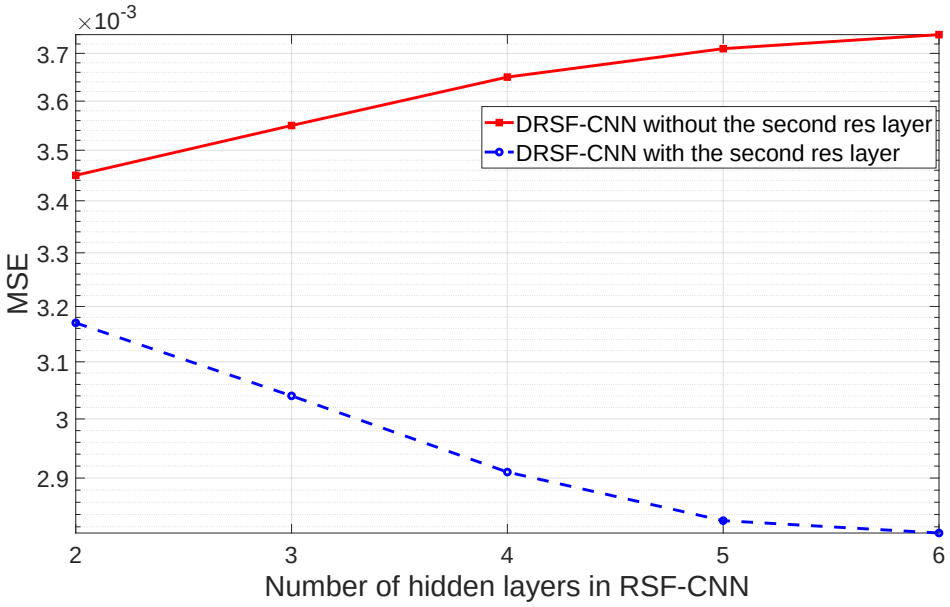


Figure 5.6: MSE of DRSF-CNN versus the number of hidden layers in RSF-CNN.

5.5.3 Convergence Analysis

The MSE performance versus the training epochs of the SF-CNN [108] and the proposed three CE networks is presented in Fig. 5.7. Note that the input of D-CNN and DRSF-CNN is the coarsely estimated CIR $\widehat{\boldsymbol{H}}_{LS}$, while the input of SF-CNN and RSF-CNN is the CFR after FFT, i.e. $\left(\widehat{\boldsymbol{H}}_{LS}\right)_{\text{zp}}\mathcal{D}$. Since the MSE of these CNNs decreases drastically at the start of training, we adopt 10 points in the first 100 epochs and 6 points in the remaining 300 epochs. In this figure, SF-CNN starts to converge after 300 epochs and has the worst performance. With the same number of layers and filters, RSF-CNN performs much better and converges faster than SF-CNN, which demonstrates the effectiveness of our residual design for frequency-domain channel estimation.

In Fig. 5.7, RSF-CNN outperforms D-CNN before 50 epochs, while D-CNN achieves better performance when these two CNNs are converged. As discussed before, the proposed D-CNN and RSF-CNN focus on different characteristics of channels in different domains, and they are trained to learn different mapping functions via

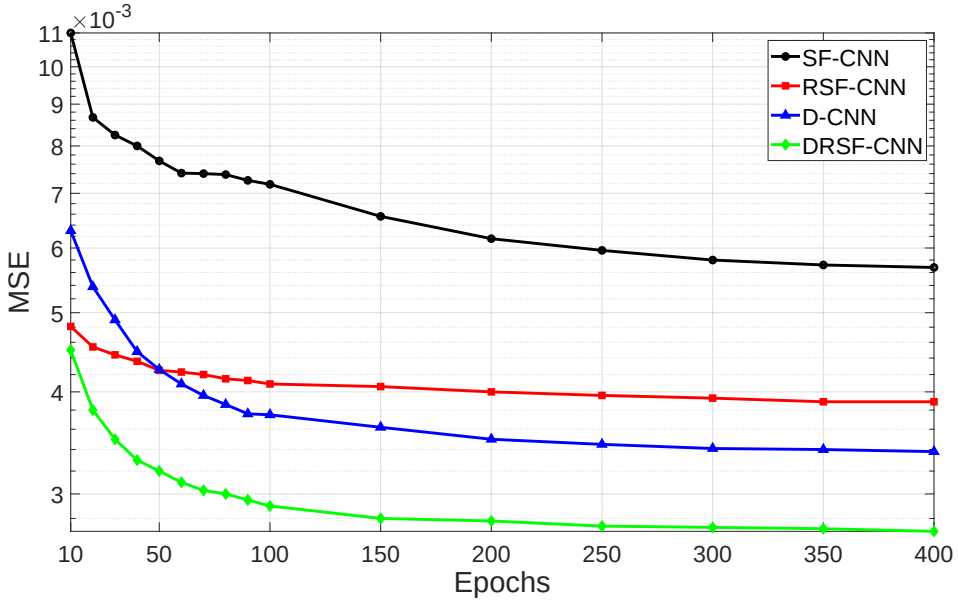


Figure 5.7: Convergence curves of the SF-CNN and the proposed CNN-based estimators under the realistic 3GPP-3D channels and 15dB SNR.

different residual layers. D-CNN aims to eliminate channel noise in the delay domain, while RSF-CNN exploits the spatial correlation between adjacent antennas and the frequency correlation of channels in different subcarriers to improve the channel estimation accuracy. Thus, by combining the efficient D-CNN and RSF-CNN in different domains, DRSF-CNN has the best performance among the four CNNs and fast convergence. More precisely, it outperforms all the converged competitors from 40 epochs and starts to converge at only 150 epochs.

5.5.4 MSE Performance and Robustness to SNRs

Fig. 5.8 illustrates the MSE Performance versus SNR of the conventional LS and LMMSE estimation, the SF-CNN, and the proposed RSF-CNN and DRSF-CNN. Without consideration of noise information, the LS has the worst estimation performance in this figure. Due to the less effect of noise, the performance gap between the LS and LMMSE becomes smaller as SNR increases. Moreover, both the SF-CNN

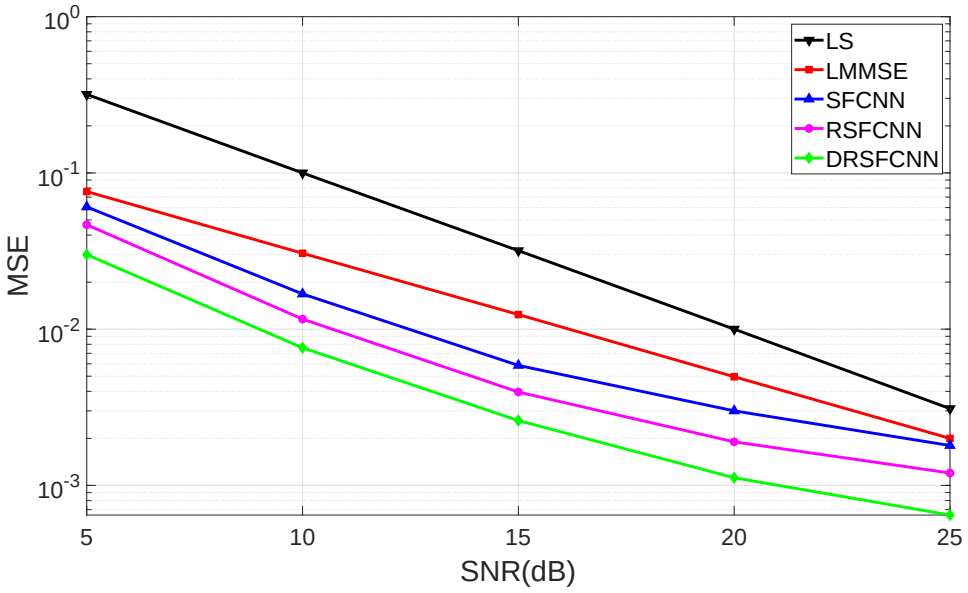


Figure 5.8: MSE versus SNR of the LS, LMMSE, SF-CNN, the proposed RSF-CNN and DRSF-CNN; Training SNR = 15dB.

and the proposed two CNNs significantly outperform the LS. Thanks to the ability to exploit channel correlations, all CNN-based schemes can efficiently refine the LS coarse estimation and improve MSE performance. With the help of the residual layer, the RSF-CNN is averagely 2 dB better than the SF-CNN in Fig. 5.8. As shown in Fig. 5.3 and Table 5.1, DRSF-CNN replaces the $2 \sim 6$th layers of RSF-CNN with a compact D-CNN and takes small-sized input. Compared to the SF-CNN and RSF-CNN with the same depth, DRSF-CNN has not only lower complexity but also better performance under all SNRs. By fusing the powerful denoiser D-CNN, the specially designed FFT layer, and the residual RSF-CNN, the DRSF-CNN can efficiently learn underlying channel features in the delay and frequency domains to minimize the overall MSE.

Unlike in [108] where different neural networks are trained separately for different SNRs, we only train one network under 15 dB SNR and test it over all operating SNRs. Thus, the robustness of the CNN-based schemes to different SNRs can also be evaluated in Fig. 5.8. Under the training SNR 15dB, it is obvious that the performance

gap between the conventional LMMSE and the three CNNs is the largest. However, for the untrained SNRs, especially when SNR = 5 dB or 25 dB, the performance gain of both SF-CNN and RSF-CNN decreases obviously. In this case, the performance advantage of DRSF-CNN over the rest two CNNs is more remarkable. Therefore, the generalization ability of DRSF-CNN to different SNRs is better than the SF-CNN [108] and the RSF-CNN which only utilize channel features in the spatial-frequency domain.

5.5.5 Detection Performance with DRSF-CNN Estimation

In the literature, most model-driven detection schemes work with perfect CSI rather than channel estimators. In Chapter 4, several state-of-the-art DU-based detection networks and the proposed MMO-Net are also evaluated with perfect CSI, which is unrealistic in practice. In [88], a joint channel estimation and signal detection architecture is proposed for a 4×4 single-user MIMO system with i.i.d. Gaussian channels. However, accurate channel estimation is much more challenging for the massive MU-MIMO-OFDM system and the 3GPP-3D channels considered in Chapter 4. In this chapter, we have proposed the DRSF-CNN, thus we are now ready to evaluate it together with the MMO-Net as an entire receiver.

In this subsection, we choose the OAMP-Net2 [88] which outperforms other detectors in Fig. 4.6-4.7 as the baseline. For the MIMO system with the DRSF-CNN channel estimator, the BER performance of two DU-based detectors is compared in Fig. 5.9. Note that the same system SNR is assumed in the pilot and data transmission stage. To the best of our knowledge, this is one of the first works that jointly evaluate the DL-based channel estimator and DU-based detectors in massive MU-MIMO-OFDM systems with real-world channels.

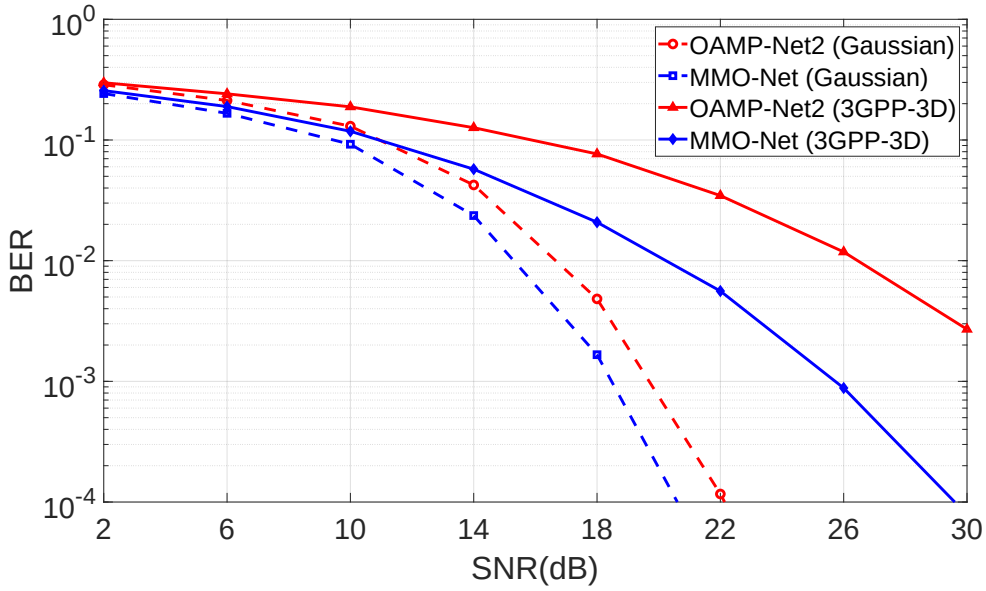


Figure 5.9: BER performance of OAMP-Net2 and MMO-Net with DRSF-CNN channel estimation for a 32×64 MIMO-OFDM system under two different channels.

In Fig. 5.9, compared to the simple case with Gaussian channels, MMO-Net has a wider performance gap with OAMP-Net2 under realistic 3GPP-3D channels. Moreover, according to the results in Fig. 4.6-4.7 and Fig. 5.9, MMO-Net suffers a smaller performance loss from channel estimation than OAMP-Net2 for both channel models, especially for 3GPP-3D channels. Specifically, if we target for BER = 10^{-2}, the performance gap between MMO-Net and OAMP-Net2 is 3.7 dB in Fig. 4.7 (a), and this gap increases to 5.1 dB in Fig. 5.9. Due to the nature of its OAMP structure, OAMP-Net2 requires strict assumptions on channel matrices. Thus, OAMP-Net2 can not generalize its excellent performance from i.i.d. Gaussian channels to realistic channels with ill conditions and spatial-frequency correlations. By contrast, MMO-Net requires no assumption on channel properties, which makes it more robust to realistic channel realizations and channel estimation errors.

5.6 Summary

Recently, deep learning-based receivers have shown promising results on simple systems in the literature. However, degraded performance or high complexity limits their universality in massive MU-MIMO-OFDM systems. This is particularly true with the conditions of a high user load and realistic channels. To solve this problem, this chapter proposes three CNN-based channel estimators.

Firstly, in Section 5.3, a frequency-orthogonal pilot scheme is developed to save time resources for massive MU-MIMO systems with high user load. This ZC-based pilot sequence is optimal for frequency-selective channel estimation in MIMO-OFDM systems. In the delay domain, a compact D-CNN is proposed to denoise coarse estimates of explicitly via a special subtractive residual layer. As a result, D-CNN outperforms the CNN-based estimators with the additive residual layer or without residual design.

In Section 5.4, to further enhance the estimation performance of D-CNN, we also propose RSF-CNN to exploit channel correlation in the spatial-frequency domain. Moreover, a customized FFT layer is added between D-CNN and RSF-CNN, resulting in the end-to-end network, DRSF-CNN. Since it can jointly learn channel features across different domains to minimize the overall MSE, DRSF-CNN achieves state-of-the-art performance and fast convergence. In addition, compared with the DNN-based CE networks and the SF-CNN [108], DRSF-CNN has lower complexity.

In Subsection 5.5.5, the competitive performance of the receiver consisting of the MMO-Net proposed in Chapter 4 and the DRSF-CNN is demonstrated under the challenging scenario, i.e., the high-loaded MIMO-OFDM system with realistic channels. However, the proposed schemes and corresponding results in Chapters 4 and 5 are obtained in infinite-precision systems. In fact, the power consumed by analog-to-

digital converters (ADCs) is dominant in modern receivers, which is a key bottleneck to wideband and massive MIMO systems. Low-resolution ADC is able to reduce power consumption significantly but leads to difficulties in accurate signal detection. Hence, in the next chapter, we will investigate the DL-based detection schemes for massive MU-MIMO-OFDM systems with low-precision quantization.

Chapter 6

Deep Unfolding-based Detection for Massive MU-MIMO-OFDM Systems with Coarse Quantization

6.1 Introduction

Massive multiple-input multiple-output (MIMO) systems equip cellular base stations (BSs) with a very large number of antennas, which can potentially provide major gains in spectral and energy efficiency [117]. However, as the number of antennas increases, the power consumption of analog-to-digital converters (ADCs) becomes very significant, which hinders the deployment of large bandwidth and large-scale antenna systems [118]. Using coarse quantization is able to provide high sampling rates at low power consumption, but signal detection becomes more challenging due to the strong nonlinearity introduced by low-resolution ADCs. Thus, there is a growing requirement to operate reliably with low-precision ADCs in massive MIMO receivers of the fifth generation (5G) communication systems, which are designed to handle wideband analog signals from multiple antennas [119].

Due to its easy integration with MIMO and low complexity, orthogonal frequency-division multiplexing (OFDM) is widely applied in communications products [120]. For frequency-selective channels considered in this chapter, we investigate the detec-

tion schemes for the uplink scenario of massive multiuser (MU) MIMO systems combined with OFDM.

6.1.1 Literature Review

Over the past decade, detection schemes for quantized MIMO systems have been extensively proposed. Linear detection algorithms like maximal ratio combining, have been studied in [121]. In [122], a belief propagation-like MIMO detector with low complexity is proposed for quantized data. Based on a minimum mean square error (MMSE) approach, the authors of [123] jointly optimize the quantizer and the iterative decision feedback equalizer for MIMO channels. Furthermore, for MU-MIMO detection tasks, a new Message Passing De-Quantization algorithm [124] is developed by simplifying Sum-Product-Algorithm. However, all of these detection algorithms have only been evaluated on frequency-flat channels. Only a few papers like [111] have applied low-resolution ADCs in the more practical case of frequency-selective channels. The results in [111] have demonstrated that systems with 4-6 bits ADC resolution can approach a similar performance to unquantized multiuser (MU)-MIMO-OFDM systems.

Deep learning (DL) has recently garnered growing interest in data detection. In [9] and [52], the entire communication system is packaged into an autoencoder (AE) in an end-to-end fashion. Different from these black-box-like receivers, our RecNet proposed in Chapter 3 replaces some function blocks in OFDM receivers with deep neural networks (DNN) and is easier to be optimized [10]. For MIMO detection, a data-driven blind receiver is proposed in [125]. Meanwhile, a model-driven detector achieves superior performance, which combines iterative approximate message passing (AMP)

architecture with online learning [90]. To avoid the extra training overhead and latency caused by online learning, a deep unfolding (DU)-based network trained offline, i.e. MMO-Net [12], is proposed for the robust detection in massive MU-MIMO-OFDM systems with realistic channels.

In recent years, DL-based detection is gaining popularity in coarsely quantized systems. In [70], an AE-based receiver is proposed for the 1-bit OFDM system. Similar to the RecNet [10], the DL-based channel estimator and signal detector are separated in 1-bit OFDM receivers [126]. With the help of block-based design, this scheme outperforms the black-box receiver in [70]. Yet these papers only consider single-antenna systems. There are also some works that apply DL tools in MIMO detection tasks with low-resolution ADCs. In [127], the authors proposed a 1-bit transceiver for small-scale MIMO, while [128] presented DNN-based massive MIMO channel estimation and detection. However, the majority of DL-based MIMO detectors with low-precision ADCs did not consider frequency-selective channels and OFDM, which is a major waveform for both uplink and downlink in 5G New Radio [120]. Reliable detection in quantized MIMO-OFDM systems is extremely difficult since the orthogonality in received OFDM signals is disrupted by the severe distortion of low-resolution ADCs.

6.1.2 Contributions

As discussed in the last subsection, both existing conventional and DL-based detection schemes hardly consider the combination of massive MU-MIMO, frequency-selective channels, OFDM, and low-precision ADCs, which is more relevant for next-generation communication systems. In addition, most NN-based works only adopt a small number of transmit antennas (like 2 or 4) and low-order modulation (up to 4-Quadrature

Amplitude Modulation (QAM)), which partly mitigate the problem of high complexity but are not common in practical scenarios. These findings motivate us to design an efficient detector for massive MU-MIMO-OFDM systems with low-precision quantization. There are two main contributions in this chapter:

1. By introducing deep-unfolding (DU) tools [129] in the detection of quantized massive MIMO-OFDM, we propose the QMMO-Net, which fuses the advantages of model-based algorithms and data-driven DNNs to achieve a good trade-off between performance and complexity. Specifically, a flexible non-linear estimator with vector learnable parameters is designed to replace the generic $\boldsymbol{x}$-update step of alternating direction method of multipliers (ADMM). This design allows QMMO-Net to handle the strong nonlinearity of low-precision ADCs.

2. Moreover, a differentiable projection is proposed to enable parameter updates via gradient descent. Compared with the generic DNN architectures in [70], [126]-[128], which have hundreds of times more trainable parameters, the specialized QMMO-Net has performance guarantees and does not require a huge number of training samples and long training time.

Our simulations demonstrate that QMMO-Net outperforms classic ADMMs and a state-of-the-art DL-based detector [90] in coarsely quantized massive MIMO-OFDM systems. Furthermore, the performance and robustness of quantized signal detection are evaluated and analyzed in high-order modulations and high-loaded scenarios (e.g., the ratio between users and the number of receiving antennas is 1/4 or 1/2), which are rarely studied in the literature.

The remainder of this chapter is organized as follows. In Section 6.2, the model of the quantized MU-MIMO-OFDM system and the quantizer are first introduced. Then, we discuss technical challenges of the massive MU-MIMO-OFDM detection with low-resolution ADCs in Section 6.3, and we also propose the corresponding specialized

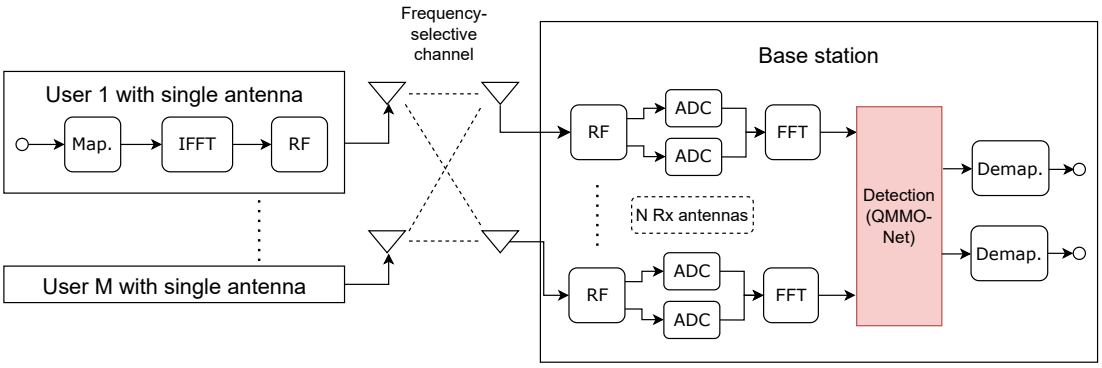


Figure 6.1: The simplified flowchart of the considered massive MU-MIMO-OFDM system. This coarsely quantized system is uncoded.

detection network, i.e., QMMO-Net. Moreover, Section 6.4 presents the simulation results to validate the performance and robustness of QMMO-Net with different user loads and modulation schemes, as well as the 1-3 bits ADCs. Finally, the conclusion of this chapter is given in Section 6.5.

6.2 Preliminaries

6.2.1 System Model

In this chapter, we consider an uplink scenario where multiple users simultaneously communicate with the BS in a single cell. Fig. 6.1 shows the massive MU-MIMO-OFDM wireless system with low-precision ADCs, including M single-antenna users, frequency-selective channels, and a base station equipped with N antennas. Note that our MU-MIMO system is uncoded (i.e. open-loop), which means the users do not require the channel state information (CSI).

On the user side, the message bits are mapped into a sequence of symbols by QAM modulation. The inverse fast Fourier transform (IFFT) is used to convert these symbols to time-domain signals, which are then transmitted through wireless channels. On the BS side, each antenna receives a noisy superposition of signals transmitted by users. For each subcarrier ω, the received symbols $\boldsymbol{y}^{\omega} \in \mathbb{C}^{N}$ can be represented as:

$$\boldsymbol{y}^{\omega} = \boldsymbol{H}^{\omega}\, \boldsymbol{x}^{\omega} + \boldsymbol{n}^{\omega} \tag{6.1}$$

where $\boldsymbol{H}^{\omega} \in \mathbb{C}^{N \times M}$ is the channel matrix, and $\boldsymbol{n}^{\omega} \in \mathbb{C}^{N}$ is the Gaussian noise vector with zero mean and variance N_0. $\boldsymbol{x}^{\omega} \in \chi$ is the transmitted symbol vector, which is chosen from a constellation set χ. To simplify the complicated notation of MU-MIMO-OFDM systems, we omit the subcarrier index ω when it is not necessary.

6.2.2 Coarse Quantization

As shown in Fig. 6.1, a pair of low-resolution ADCs is utilized to quantize the real and imaginary parts of each radio-frequency (RF) chain's outputs at the BS side. Specifically, the complex-valued element of the time-domain received signal vector, i.e., $s \in \mathbb{C}$, is first converted to real scalars $s_R, s_I \in \mathbb{R}$ and then quantized via $\mathcal{Q}(s_R, s_I)$. The mapping function $\mathcal{Q}(\cdot) : \mathbb{R} \to \mathcal{M}$ is a quantizer, where $\mathcal{M}$ denotes the finite quantization alphabet. In what follows, the quantization operation $\mathcal{Q}(\cdot)$ will be frequently applied element-wise to vectors and matrices.

For low-order modulation schemes like the 4-QAM widely used in [70], [126]-[128], 1-bit quantization is considered. For the nth BS antenna, the time-domain received signal $\widetilde{\boldsymbol{y}}_n$ at the output of RF chain is quantized by a pair of 1-bit ADCs as:

$$\boldsymbol{y}_q = \frac{1}{\sqrt{2}} \operatorname{sign}\left(\Re\left(\widetilde{\boldsymbol{y}}_n\right)\right) + \frac{j}{\sqrt{2}} \operatorname{sign}\left(\Im\left(\widetilde{\boldsymbol{y}}_n\right)\right). \tag{6.2}$$

where sign$(\cdot)$ indicates the 1-bit quantizer with sign$(z) = 1$ and sign$(-z) = -1$ for $z \geq 0$. For the higher-order QAM modulations, we utilize the B-bit scalar quantizer, i.e. $\mathcal{Q}(s_R) = q_i$, where q_i is the quantization label chosen from $\mathcal{M} \in \{q_1, q_2, \ldots, q_{2^B}\}$. Note that the real part s_R and the imaginary part s_I share the same $\mathcal{M}$. The quantized value of s_R or s_I, i.e. q_i, is determined by $b_{i-1} \leq s_R < b_i$, where b_i is the i-th quantization bin boundary according to $b_0 = -\infty < b_1 < \ldots < b_{2^B} = +\infty$. Therefore, the B-bit quantization output of the received signal $\widetilde{\boldsymbol{y}}_n$ is expressed as:

$$\boldsymbol{y}_q = \mathcal{Q}\left(\Re\left(\widetilde{\boldsymbol{y}}_n\right)\right) + j\mathcal{Q}\left(\Im\left(\widetilde{\boldsymbol{y}}_n\right)\right) \tag{6.3}$$

After the quantization operation, the $\boldsymbol{y}_q$ is processed by W-point fast Fourier transform (FFT), where W is the number of subcarriers in one OFDM symbol. Next, the frequency-domain received vectors, channel matrices, and the training labels $\boldsymbol{x}$ are reorganized in the real-value form following the equation (4.2) and then input into the proposed detection network.

6.3 QMMO-NET: A Detection Network for Quantized MU-MIMO-OFDM

In this section, we first discuss the difficulties of the massive MU-MIMO-OFDM detection with low-precision ADCs. Then we present the corresponding specialized architecture and the DU-based detection scheme that unfolds the iterations of our model-driven algorithm into a layer-wise neural network.

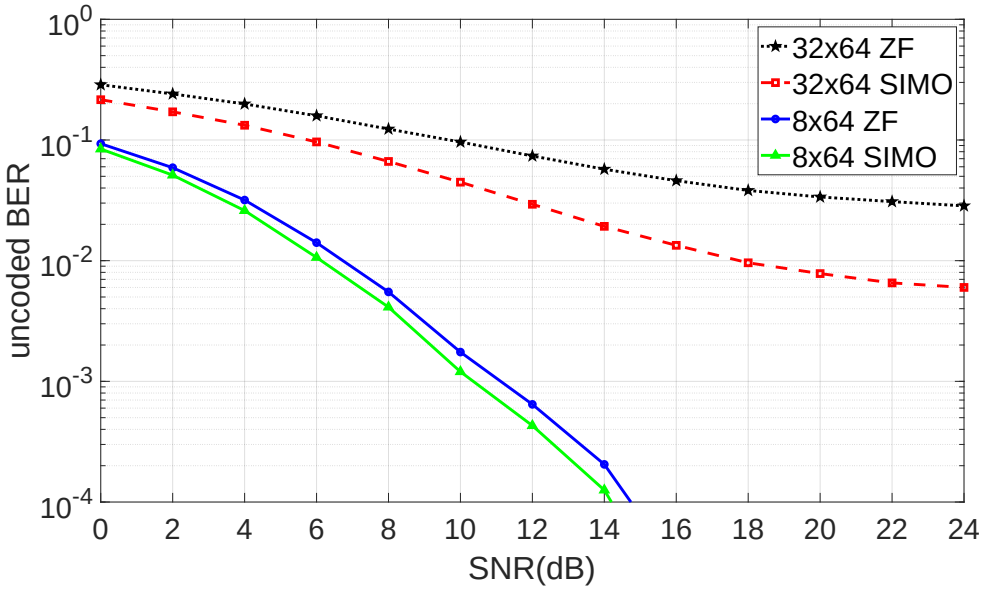


Figure 6.2: BER versus SNR curves of the 32×64 and 8×64 uncoded MU-MIMO-OFDM systems with 16-QAM and 3-bit ADCs.

6.3.1 Technical Challenges

Although preprocessing data with the help of CSI can mitigate the interference between transmitted signals, the overhead of obtaining CSI at the transmitter side can be significant, especially for large-scale MIMO systems. Thus, we consider an open-loop MU-MIMO-OFDM system in this chapter, which means all burdens of data processing like handling the co-channel interference (CCI) are placed on the receiver side. Moreover, the explosion of advanced applications has propelled the development of wireless communication into 5G to achieve a thousandfold capacity and massive connectivity. Therefore, for BSs with a fixed number of antennas, it is practically important to serve as many users as possible. Whereas, the CCI notably increases when the ratio between users and the number of receiving antennas becomes larger. Fig. 6.2 depicts the bit error rate (BER) versus signal-to-noise ratio (SNR) of our uncoded MU-MIMO-OFDM system with different user loads and 3-bit ADCs. Compared with that in the 32×64 high-loaded case, the performance gap between the

zero-forcing (ZF) detection and the SIMO lower bound without CCI is significantly diminished in the 8×64 case ($M \times N$ denotes the system with M users and N receiving antennas). Therefore, to get an acceptable BER performance in MIMO systems with low-resolution ADCs, prior works mainly adopt low-loaded settings of Rx antennas with minor CCI, e.g., 4 Tx antennas with 64 Rx antennas in [128].

Although the MIMO-OFDM system has many advantages and has become the main scheme of 5G new ratio, it is more sensitive to low-resolution ADCs than the single-carrier MIMO systems considered in the literature. This is because the orthogonality in the received signals in the frequency domain is no longer preserved due to the severe inter-carrier interference (ICI) caused by coarse quantization in the time domain [70]. The disruption of orthogonality and the severe distortion of low-precision ADCs make the accurate detection for quantized MU-MIMO-OFDM systems very challenging.

With the ability to deal with non-linear distortions, conventional DNN architectures show promising performance in either coarsely quantized MIMO [127]-[128] or OFDM systems [70], [126]. However, due to the data-driven nature, the huge parameter sizes of these black-box DNNs will further increase in massive MU-MIMO-OFDM systems. For example, for the single-antenna OFDM in [126], there are at least $256 \times 256 = 65536$ parameters in the first fully-connected layer when the input size is 256, and the network needs 10000 epochs to be well trained. When combined with 32×64 MIMO chains in our system, the parameter number of the DNN-based scheme [126] will increase by at least two orders of magnitude. Correspondingly, the training samples and time will also significantly increase. Thus, the substantial overhead hinders the implementation of generic DNNs in massive MU-MIMO-OFDM systems. Moreover, such DNNs are not interpretable and highly rely on the training data that must have similar statistics as the data from where they operate, which may lead to significant performance reduction of these DNNs in dynamic communication systems.

In summary, in the context of high-loaded massive MU-MIMO-OFDM systems with low-resolution ADCs, it is desirable to develop a low-complexity detection scheme that can cope with severe CCI and non-linear distortions. This is the main aim of this chapter, and the scheme we proposed will be introduced in the next subsection.

6.3.2 QMMO-Net Design Based on Deep Unfolding

An efficient detector is supposed to approach a good trade-off between performance and complexity. Unlike the typical DNN-based architectures, model-driven networks are interpretable and less complex in massive MU-MIMO-OFDM systems. In consideration of the success of DU-based detectors in our work [12] and the literature [86]-[89], here we propose to extend the DU-based schemes under infinite-precision quantization to the systems with low-precision quantization.

According to the numerical results in Chapter 4, the ADMM-based architecture of the proposed MMO-Net has superior performance and robustness in massive MU-MIMO-OFDM systems with high user load. Therefore, we first revisit the ADMM-based MIMO detection algorithm derived in Subsection 4.3.2 as follows:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \left\{ \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{x}-\boldsymbol{z}_k+\boldsymbol{u}_k\|_2^2 \right\} \\ &= \left(\boldsymbol{H}^H\boldsymbol{H}+\rho\boldsymbol{I}\right)^{-1}\left[\boldsymbol{H}^H\boldsymbol{y}+\rho\left(\boldsymbol{z}_k-\boldsymbol{u}_k\right)\right] \end{aligned} \tag{6.4}$$

$$\begin{aligned} \boldsymbol{z}_{k+1} &= \arg\min_{\boldsymbol{z}} \left\{ g(\boldsymbol{z}) + \frac{\rho}{2}\|(\boldsymbol{x}_{k+1}+\boldsymbol{\mu}_k)-\boldsymbol{z}\|_2^2 \right\} \\ &= \prod\nolimits_{\chi}(\boldsymbol{x}_{k+1}+\boldsymbol{u}_k) \end{aligned} \tag{6.5}$$

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \boldsymbol{z}_{k+1} + \boldsymbol{x}_{k+1} \tag{6.6}$$

where $\boldsymbol{u}_k$ is the scaled dual variable, k is the iteration index, ρ is the step size of ADMM, and $\boldsymbol{I}$ is a unitary matrix that has the same size as $\boldsymbol{H}^H\boldsymbol{H}$. As $g(\boldsymbol{z})$ is the indicator function of the constellation set χ, the $\boldsymbol{z}$-update can be solved by $\prod_{\chi}$, i.e. a Euclidean projection onto χ [92].

Here, the $\boldsymbol{x}$-update (6.4) derived based on the derivative of equation (4.13) involves minimizing $\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|_2^2$, which only considers the linear system model (6.1). As discussed in Subsection 6.3.1, low-precision quantization in (6.2)-(6.3) introduce severe nonlinearity in the massive MU-MIMO-OFDM systems, which makes the detection task extremely difficult. Due to its very limited flexibility, the linear x-update in (6.4) can not cope with the strong non-linear distortions in such a challenging system. On the other hand, limited by the unacceptable complexity and training overhead, the overly general DNN models in [70], [126]-[128] are also unrealistic. Therefore, a specialized architecture with moderate flexibility and non-linear functionality is necessary.

Similar to the problem of the massive MIMO detection with infinite-precision quantization in Chapter 4, the coarsely quantized detection in this chapter requires us to solve high-dimensional convex optimization problems, which is also one of the main interests in the machine learning field. The form of a generic convex optimization problem can be expressed as:

$$\text{minimize} \quad f(\boldsymbol{x})+g(\boldsymbol{z}), \quad \text{subject to} \quad \boldsymbol{x}=\boldsymbol{z} \tag{6.7}$$

where $f(\boldsymbol{x})=\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|_2^2$ is the estimation loss for the MIMO detection task in (4.3). Except for the ADMM algorithm in (6.4)-(6.6), the forward-backward splitting (FBS) method [130] can also solve this optimization problem efficiently. In [111], an FBS-

based detector is used for the coded MU-MIMO-OFDM system with quantization:

$$\boldsymbol{x}_{k+1} = \text{prox}\left(\boldsymbol{x}_k - \tau_k \boldsymbol{A}^H \nabla h\left(\boldsymbol{A}^H \boldsymbol{x}_k\right)\right) \tag{6.8}$$

where prox() denotes the non-linear proximal operator. For $\boldsymbol{a} = \Re(\widetilde{\boldsymbol{y}})$, $\nabla h()$ is the gradient of the negative log-likelihood function $-\log p(\boldsymbol{q} \mid \boldsymbol{a})$, which is calculated as follows:

$$[\nabla h(\mathbf{a})]_i = \frac{\exp\left(-\frac{(u(q_i)-a_i)^2}{2\sigma^2}\right) - \exp\left(-\frac{(\ell(q_i)-a_i)^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2}\left[\Phi\left(\frac{u(q_i)-a_i}{\sigma}\right) - \Phi\left(\frac{\ell(q_i)-a_i}{\sigma}\right)\right]} \tag{6.9}$$

where $u(q_i) = b_i$ and $\ell(q_i) = b_{i-1}$ are the upper and lower quantization boundary of q_i, respectively. $\Phi(\cdot)$ denotes the the cumulative distribution function. It is obvious that the computation of $\nabla h()$ with exponential functions and cumulative distribution functions in [111] is highly computationally intensive. To avoid computing the overly complex gradient $\nabla h()$ (6.9), we derive a new $\boldsymbol{x}$-update in an iterative soft-thresholding algorithm (ISTA) fashion [131] as:

$$\boldsymbol{x}_{k+1} = \text{prox}\left(\boldsymbol{x}_k - \rho_k \boldsymbol{H}^H\left(\boldsymbol{H}\boldsymbol{x}_k - \boldsymbol{y}\right)\right) \tag{6.10}$$

where ρ_k is the step size. Here, $\boldsymbol{H}^H\left(\boldsymbol{H}\boldsymbol{x}_k - \boldsymbol{y}\right)$ is the gradient (partial derivative) of $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2$ over $\boldsymbol{x}$ in (6.7).

Like that in the $\boldsymbol{x}$-update (6.4), a scalar step size ρ_k in (6.10) is not sufficient to correct the high-dimensional gradients in our coarse quantized system with strong nonlinearity properly. Generally, with the help of the trainable weight matrices and bias vectors, DNNs can outperform linear algorithms like minimum mean squared error (MMSE) in non-linear systems. Moreover, unlike the conventional analytical or heuristic selection strategies, deep unfolding is able to optimize the high-dimensional parameters jointly

in model-based algorithms. Inspired by these findings, we upgrade the scalar ρ_k to a trainable vector $\boldsymbol{\rho}_k$ and introduce an additive gradient correction $\boldsymbol{c}_k$ in the $\boldsymbol{x}$-update (6.10). The resulting new $\boldsymbol{x}$-update is given by:

$$\boldsymbol{x}_{k+1} = \text{prox}\left(\boldsymbol{x}_k - \boldsymbol{\rho}_k \odot \boldsymbol{H}^H\left(\boldsymbol{H}\boldsymbol{x}_k - \boldsymbol{y} + \boldsymbol{c}_k\right)\right) \tag{6.11}$$

where $\odot$ is the Hadamard product (also known as element-wise product). Due to the non-linear operation $\odot$, $\boldsymbol{\rho}_k \odot \boldsymbol{H}^H\left(\boldsymbol{H}\boldsymbol{x}_k - \boldsymbol{y} + \boldsymbol{c}_k\right)$, is already a non-linear gradient update before the prox() operation, which is different from that in (6.10). Compared with the generic $\boldsymbol{x}$-update of ADMM in (6.4), the function (6.11), which is specially designed to handle the non-linear distortions of coarse quantization, is more flexible. By combining the specialized $\boldsymbol{x}$-update (6.11) with the ADMM iterations (6.5)-(6.6), we are now ready to present the QMMO-Net in Algorithm 2:

Algorithm 2 QMMO-Net for MU-MIMO-OFDM Detection with Coarse Quantization

Preprocessing: $\boldsymbol{x}_{MMSE} = \left(\boldsymbol{H}^H\boldsymbol{H} + \sigma\boldsymbol{I}\right)^{-1}\boldsymbol{H}^H\boldsymbol{y}$;

Transform $\boldsymbol{H}$, $\boldsymbol{y}$ and $\boldsymbol{x}_{MMSE}$ into the real form following Eq. (4.2)

Input: $\boldsymbol{H}$, $\boldsymbol{y}$ and $\boldsymbol{x}_{MMSE}$

Initialization: $\boldsymbol{x}_1 = \text{prox}\left(\boldsymbol{x}_{MMSE}\right)$; $\quad \boldsymbol{z}_0 = 0$; $\quad \boldsymbol{u}_0 = 0$

for $\omega = 1 : W$ **do**

  for $k = 0 : (T-1)$ **do**

    $\boldsymbol{x}_{k+1}^{\omega} = \text{prox}\left(\boldsymbol{x}_k^{\omega} - \boldsymbol{\rho}_k^{\omega} \odot \left(\boldsymbol{H}^{\omega}\right)^H\left(\boldsymbol{H}^{\omega}\boldsymbol{x}_k^{\omega} - \boldsymbol{y}^{\omega} + \alpha_k^{\omega}\left(\boldsymbol{u}_k^{\omega} - \boldsymbol{z}_k^{\omega}\right)\right)\right)$

    $\boldsymbol{z}_{k+1}^{\omega} = \varepsilon_k^{\omega}\left(\boldsymbol{H}^{\omega}\boldsymbol{x}_{k+1}^{\omega} - \boldsymbol{y}^{\omega} + \boldsymbol{u}_k^{\omega}\right)$

    $\boldsymbol{u}_{k+1}^{\omega} = \boldsymbol{u}_k^{\omega} + \theta_k^{\omega}\left(\boldsymbol{H}^{\omega}\boldsymbol{x}_{k+1}^{\omega} - \boldsymbol{y}^{\omega} - \boldsymbol{z}_{k+1}^{\omega}\right)$

  end for

end for

Output: $\widehat{\boldsymbol{x}}_T$

In Algorithm 2, k and ω are the layer and subcarrier index, T and W are the number of layers and subcarriers in an OFDM symbol, separately. Each layer of QMMO-Net corresponds to an iteration of the model-based architecture. After the preprocessing in the complex domain, all the inputs are transformed into the real domain following (4.2). In the first layer ($k=0$), $\boldsymbol{x}_{k+1}^{\omega} = \text{prox}\left(\boldsymbol{x}_{MMSE}^{\omega}\right)$. To update the additive gradient correction $\boldsymbol{c}_k$ in the $\boldsymbol{x}$-update (6.11) iteratively between the layers of QMMO-Net, we rewrite $\boldsymbol{c}_k$ as $\alpha_k^{\omega}\left(\boldsymbol{u}_k^{\omega} - \boldsymbol{z}_k^{\omega}\right)$, where α_k^{ω} is a trainable scalar. Note that the size of $\boldsymbol{z}_k$ and $\boldsymbol{u}_k$ is M as $\boldsymbol{x}_k$ in (6.4), but the size of $\boldsymbol{z}_k^{\omega}$ and $\boldsymbol{u}_k^{\omega}$ is N in Algorithm 2. Correspondingly, all the $\boldsymbol{x}_{k+1}^{\omega}$ in the $\boldsymbol{z}$- and $\boldsymbol{u}$-updates (6.5)-(6.6) are replaced by the residual error $\left(\boldsymbol{H}^{\omega}\boldsymbol{x}_{k+1}^{\omega} - \boldsymbol{y}^{\omega}\right)$ in Algorithm 2 to ensure the consistency of the variable size.

Intuitively, adding proper learnable parameters into the QMMO-Net architecture can increase its flexibility. Since the proximal operator prox() in the $\boldsymbol{x}$-update is equivalent to a projection function, the $\prod_{\chi}$ in (6.5) is replaced by a new trainable scalar ε_k^{ω} in the $\boldsymbol{z}$-update. In addition, an extra trainable parameter θ_k^{ω} is added to the $\boldsymbol{u}$-update. Hence, in QMMO-Net, $\left\{\left\{\rho_k^{\omega}, \alpha_k^{\omega}, \varepsilon_k^{\omega}, \theta_k^{\omega}, \lambda_k^{\omega}, b_k^{\omega}\right\}_{k=0}^{T-1}\right\}_{\omega=1}^{W}$ is the set of learnable parameters, where λ_k^{ω} and b_k^{ω} are in the prox().

For the MIMO detection problem with the constellation set χ, the proximal operator prox() in Algorithm 2 can be viewed as a projection onto χ. Such traditional projections are generally not differentiable, which hinders the stochastic gradient descent-based optimization in QMMO-Net. In [86], a multilevel projection with a sum of non-linear activations is proposed for high-order modulations. As a popular non-linear activation for neural networks, the hyperbolic tangent tanh() generally converges faster than the sigmoid used in [86] due to its bigger gradients. Besides, the zero-centered nature of tanh() is more suitable for the constellation set of QAM modulation, e.g. $\{-3,-1,1,3\}$ for both the real and imaginary parts of 16-QAM. Hence, in Subsection 4.4.3, we

propose a general tanh-based projection (4.21). Here, instead of using fixed τ_i like in (4.22), we add one more trainable parameter b_k^{ω} for the quantized MU-MIMO-OFDM detection task in this chapter, resulting in a differentiable non-linear proximal operator as follows:

$$\text{prox}(\boldsymbol{x}) = \tanh(\lambda \boldsymbol{x} - b) + \tanh(\lambda \boldsymbol{x}) + \tanh(\lambda \boldsymbol{x} + b) \tag{6.12}$$

6.4 Simulation Results

In this section, we first introduce system setup and training specifications. Then, we numerically evaluate the proposed QMMO-Net and compare its detection performance with both the conventional and DU-based detection schemes in the proposed massive MU-MIMO-OFDM system with low-precision quantization.

6.4.1 Implementation Details

The multipath Rayleigh fading channel is used in our experiments. The entries of the channel matrices are independently sampled from a zero-mean i.i.d Gaussian distribution with variance $(1/N)$, and the number of channel taps is 8. Note that $M \times N$ MIMO channels denote the system with M users with a single antenna and N receive antennas. In addition, there are 64 subcarriers in an OFDM symbol, and the cyclic prefix length is 16. Thus, the total size of each complex channel matrix $\boldsymbol{H}$ is $64 \times N \times M$. Different from the 4-QAM modulation widely used in [70], [126]-[128], a more challenging 16-QAM constellation is considered in this chapter. For the quantizer designed for high-order modulation schemes in (6.3), the quantization bin

boundaries $\{b_i\}$ and quantization alphabet $\mathcal{M}$ are generated based on the Lloyd-Max method [132]. In our simulation, two system settings are considered, i.e., a 16×64 MIMO system with 16-QAM and 2-bit ADCs, and a 32×64 MIMO system with 16-QAM and 3-bit ADCs.

The parameter selections of conventional ADMM algorithms are suggested by some prior papers such as [91], which may be helpful in finding good initial values of trainable parameters for ADMM-based detection networks like the MMO-Net proposed in Chapter 4. However, with a specialized $\boldsymbol{x}$-update (6.11) for quantized detection and more parameters including a vectorial one, the proposed QMMO-Net is quite different from the previous ADMM-based architectures. Thus, we empirically initialize the trainable scalar parameters $\{\alpha, \varepsilon, \theta, \lambda, b\}$ as $\{1, 0.1, 0.5, 1, 2\}$. Each element of the trainable vector $\boldsymbol{\rho}$ is randomly initialized between [0.1, 0.2]. To further improve the detection performance for coarsely quantized OFDM systems with non-orthogonal subcarriers, we train different sets of learnable parameters for each subcarrier.

All DL-based detectors in the simulations are trained in 300 epochs offline by 6000 packets of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{H})$ and tested by 1000 packets per SNR that are independent of the training dataset. Here, $\boldsymbol{x}$ is the training label, and each packet contains one OFDM symbol. In our training trials, the Adam optimizer with an exponential decay of initial learning rate (LR) outperforms a fixed initial LR. The initial LR of the exponential decay is 0.0004, and the corresponding batch size is 400. Moreover, we adopt a common mean squared error (MSE) loss. All the operations before detection, including the channel generation, the signal processing blocks in our coarsely quantized MU-MIMO-OFDM system (Fig. 6.1) and data preprocessing in Algorithm 2, are implemented with complex-valued data in Matlab. Then, all the detectors are trained and evaluated in Python with Tensorflow backend for a fair comparison.

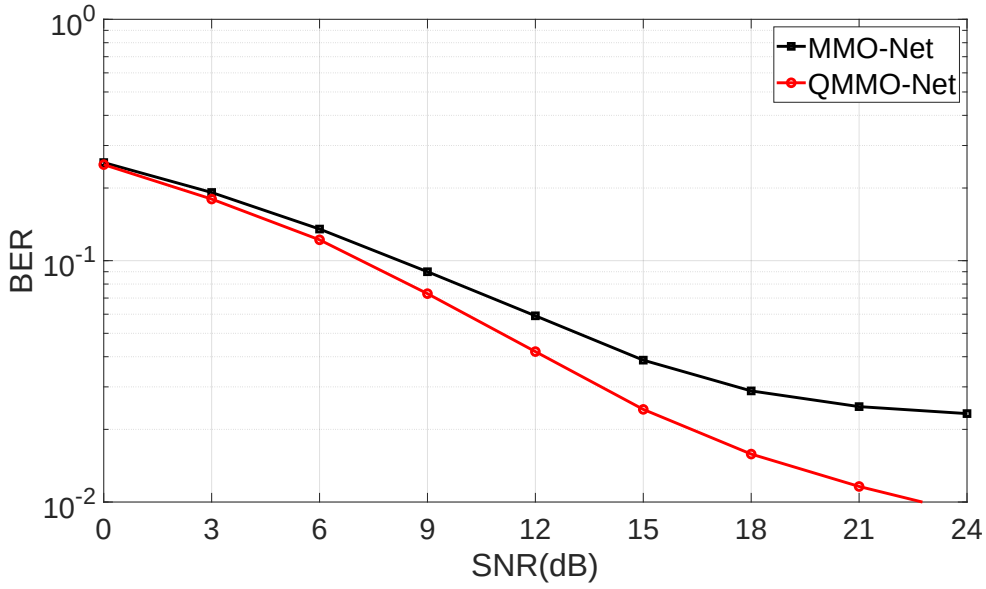


Figure 6.3: BER versus SNR curves of MMO-Net and QMMO-Net for 32×64 MU-MIMO-OFDM system with 16-QAM and 3-bit ADCs.

6.4.2 Impact of Network Architecture and Layer Number

To demonstrate the superiority of the novel architecture proposed in this chapter, we compare the BER performance of the MMO-Net (Algorithm 1) and the QMMO-Net specialized for quantized MIMO-OFDM systems in Fig. 6.3. Here, we choose a massive MU-MIMO-OFDM system with high user load (32×64), where MMO-Net demonstrates its superior performance in Chapter 4. To get an acceptable BER in the uncoded and high-loaded MIMO system, 3-bit ADCs are used. Firstly, compared to that in the infinite-precision 32×64 MIMO-OFDM system in Fig. 4.6 (a), MMO-Net has a significant performance reduction in Fig. 6.3, especially for 10 dB or higher SNR. Although MMO-Net is good at dealing with serious CCI caused by the high user load, its linear $\boldsymbol{x}$-update with only a scalar step size can not properly correct the received symbols in coarsely quantized MIMO-OFDM systems with severe ICI and nonlinearity. By contrast, with the help of the novel network skeleton in Algorithm 2,

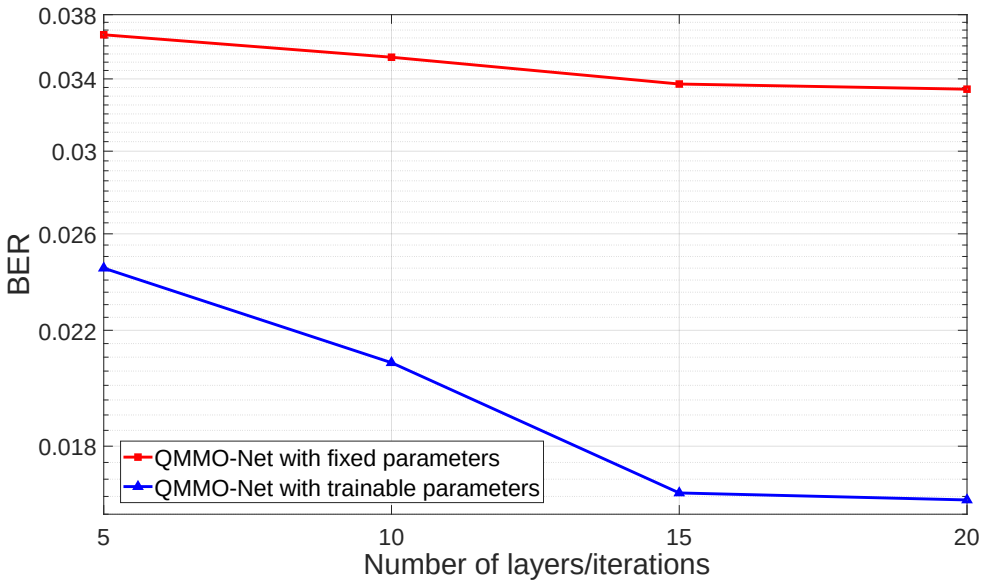


Figure 6.4: The impact of the number of layers or iterations for different QMMO-Net (16×64 MU-MIMO-OFDM system with 16-QAM and 2-bit ADCs; SNR = 18 dB).

including the non-linear $\boldsymbol{x}$-update (6.11) with a trainable vector and the corresponding $\boldsymbol{z}$- and $\boldsymbol{u}$-updates, QMMO-Net outperforms MMO-Net for quantized MIMO-OFDM detection tasks. In addition, the performance gap between MMO-Net and QMMO-Net becomes larger as the SNR increases.

In Fig. 6.4, the BER performance versus the layer number of QMMO-Net is illustrated to show the impact of network depth on detection performance. Note that the results are obtained in a 16×64 MU-MIMO-OFDM system with 16-QAM and 2-bit ADCs. The two QMMO-Nets in Fig. 6.4 have the same architecture and parameters (Algorithm 2), the only difference between them is with or without the training process. First, from the figure, it is obvious that the QMMO-Net with trainable parameters outperforms the untrained one. This result demonstrates that the jointly learned parameters are optimal or near-optimal and can improve the performance of model-based detection algorithms in MIMO-OFDM systems with low-precision ADCs. Furthermore, since the algorithm parameters of untrained QMMO-Net remain unchanged between iterations, its BER

decreases relatively slowly as the increase of iteration number. In contrast, the layer-wise trainable parameter set can help QMMO-Net obtain performance gains from increased depth. However, the performance improvement is minor when the number of layers is larger than 15. Thus, we set the number of layers for QMMO-Net to $T = 15$ in the following simulations.

6.4.3 Detection Performance

Compared to the end-to-end AE-based OFDM detector [70], an improved one-bit receiver based on model-aided architecture has better BER performance [126]. Thus, we tried to apply this DNN-based receiver in our system as a benchmark, but it did not converge to a stable solution with our training set. This is because the scheme in [126] is still based on two generic DNNs with fully-connected layers. For our massive MU-MIMO-OFDM system, there are about $4M^2N^2TW$ trainable parameters in this improved DNN-based receiver, while the proposed QMMO-Net only has $(M+5)TW$ parameters. Therefore, in coarsely quantized MU-MIMO-OFDM systems, we compare the BER performance of the following detectors:

- Unquan. MMSE: A widely-used linear MMSE detector with infinite-precision ADCs used as the lower bound of BER performance.
- MMSE: A linear MMSE with low-resolution ADCs.
- ADMM: A MIMO detector based on a generic ADMM architecture with only one scalar parameter, as shown in (6.4)-(6.6).
- MMNet-V: A model-driven detection network with a linear estimation step and a non-linear denoiser, which outperforms state-of-the-art detectors [90]. Similar to QMMO-Net, we upgrade its learnable step size θ_t of MMNet to a vector $\boldsymbol{\theta}_t$ for a fair comparison.

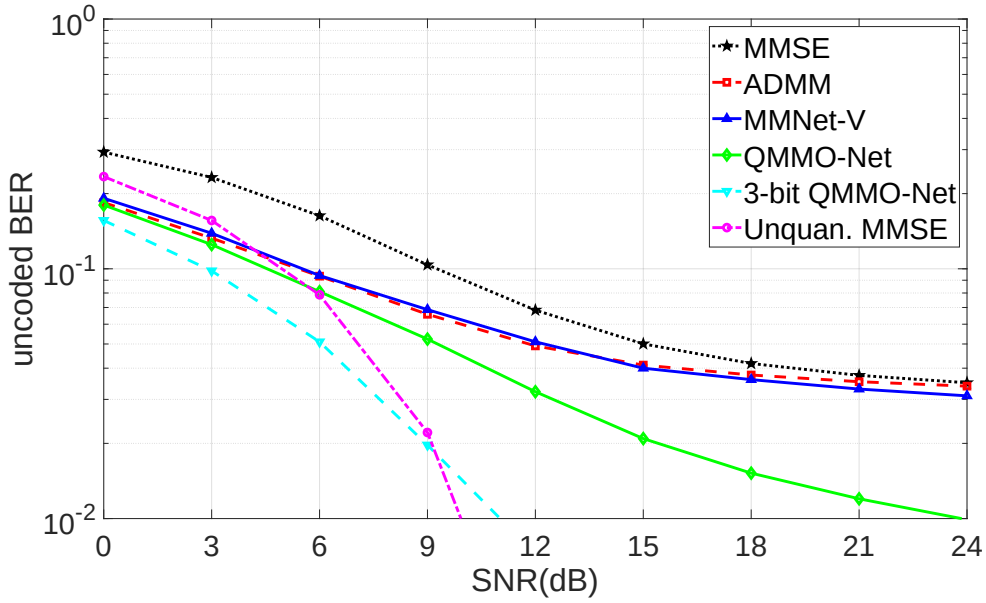


Figure 6.5: BER versus SNR curves for the uncoded 16×64 MU-MIMO-OFDM system with 16-QAM and 2-bit ADCs.

- QMMO-Net: The proposed DU-based detection network specialized for coarsely quantized massive MU-MIMO-OFDM systems, as described in Subsection 6.3.2.

The performance of all detection algorithms is first evaluated in 16×64 MIMO system with 2-bit ADCs, as shown in Fig. 6.5. For the SNRs below 15 dB, ADMM provides competitive performance compared to MMSE. Thanks to the learnable vectorial parameter $\boldsymbol{\theta}_t$, MMNet-V outperforms ADMM at SNR $>$ 15 dB. QMMO-Net has a much lower BER than MMNet-V, especially under high SNRs. As described in [90], the non-linear denoiser of MMNet is optimized for Gaussian noise. However, with 2-bit ADCs, the linear $\boldsymbol{x}$-update of MMNet-V can no longer provide a nearly Gaussian estimation at the input of denoiser like in infinite-precision systems. In contrast, the adaptive prox() in (6.12) does not rely on any special conditions. Moreover, except for the multiplicative correction $\boldsymbol{\rho}_k^{\omega}$ like in MMNet-V, QMMO-Net also has an additive gradient correction $\boldsymbol{\alpha}_k^{\omega}\left(\boldsymbol{u}_k^{\omega}-\boldsymbol{z}_k^{\omega}\right)$. Due to these features, in the extreme scenario

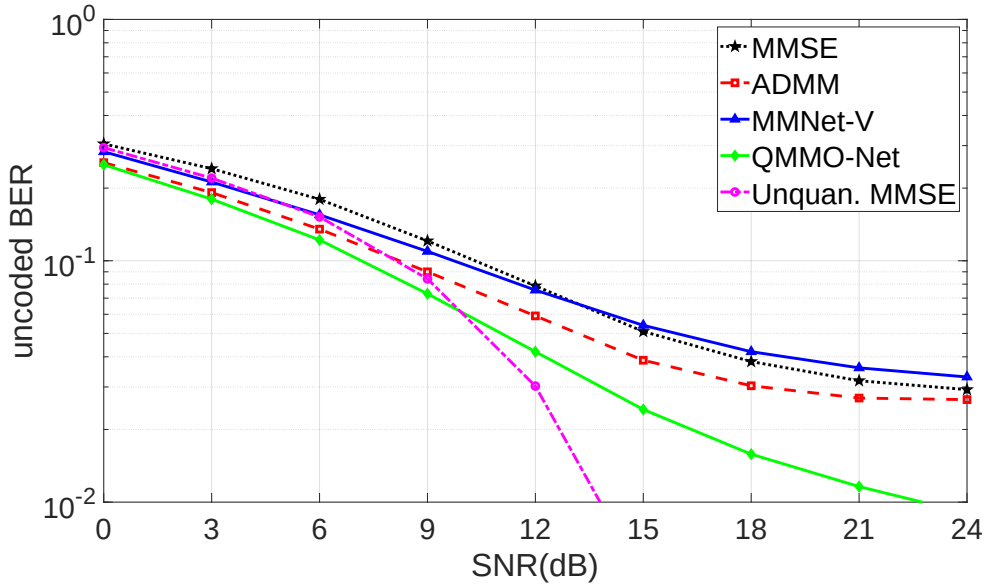


Figure 6.6: BER versus SNR curves for the uncoded 32×64 MU-MIMO-OFDM system with 16-QAM and 3-bit ADCs.

with severe CCI and ICI from very low-resolution ADCs, QMMO-Net has a 3 dB gain to ADMM at BER of 0.05 and can still achieve an acceptable BER (10^{-2}) for uncoded systems. In addition, to show the serious distortion caused by 2-bit ADCs in the 16-QAM MIMO-OFDM system, we also present the BER performance of the 3-bit QMMO-Net in Fig. 6.5. The QMMO-Net with 3-bit ADCs performs much better than that with 2-bit ADCs and even better than the unquantized MMSE when SNR is less than 9 dB.

Fig. 6.6 illustrates the BER versus SNR curves of the above detectors in a 32×64 MIMO system. As discussed in Subsection 6.3.1, compared with the settings in the literature of quantized detection, 32×64 MIMO is a very high-loaded case, so here we use 3-bit ADCs. In this case, the serious CCI caused by large user-to-BS-antenna ratios is the dominant source of error. Due to its robustness to the high load of BS antennas, ADMM has better BER performance than the trainable MMNet-V based on the AMP architecture over all SNRs. In Fig. 6.6, QMMO-Net shows state-of-the-

art detection performance, which is superior to the benchmarks. It even outperforms Unquan. MMSE under 10 dB or lower SNRs. Compared with 2-bit ADCs in Fig. 6.5, the QMMO-Net with higher CCI and 3-bit ADCs has worse BER when SNR is less than 18dB, while it performs better if SNR $>$ 18 dB. This is because low-precision ADCs fundamentally have poor performance at medium and high SNRs [70]. This is also why the BER curves of MMSE, ADMM, and MMNet-V are almost flat in both Fig. 6.5 and 6.6 when SNR $>$ 21 dB. However, with the help of the specialized non-linear $\boldsymbol{x}$-update, the adaptive projection, and the flexible architecture with high-dimensional learnable parameters optimized by DU tools, QMMO-Net still has potential to achieve lower BERs at higher SNRs, as shown in Fig. 6.5-6.6.

6.5 Summary

In Chapter 4, we investigate the detection performance in massive MU-MIMO-OFDM systems with high user load and propose the MMO-Net that achieves a good trade-off between BER performance and network complexity. In this chapter, we extend our research work to the systems with low-resolution ADCs. Firstly, the system model and the quantizer for 16-QAM and higher-order modulation schemes are introduced in Section 6.2.

Section 6.3.1 explores the difficulties of massive multiuser MIMO-OFDM detection with low-precision quantization. To solve these problems, we propose QMMO-Net in Subsection 6.3.2, a novel deep unfolding-based detection scheme that fuses the architecture specialized for quantized MIMO-OFDM detection with data-driven techniques. Specifically, to handle the severe distortions from coarse quantization, we

derive a non-linear $\boldsymbol{x}$-update with both multiplicative and additive gradient corrections. We also add multiple trainable parameters to increase the model flexibility. With the help of the proposed differentiable proximal operator and DU tools, these parameters including a high-dimensional step size $\boldsymbol{\rho}$ can be jointly optimized.

In Section 6.4, simulation results demonstrate that QMMO-Net outperforms traditional detection algorithms and DU-based detection networks in coarsely quantized MU-MIMO-OFDM systems. By combining the power of domain knowledge with data, our QMMO-Net has strong robustness to the non-linear effects of coarse quantization and the co-channel interference caused in high user load scenarios.

Chapter 7

Conclusions and Future Work

This thesis has contributed to the field of advanced receiver algorithms with the application of deep learning (DL) techniques for massive multiuser MIMO-OFDM systems in 5G and beyond. Data-driven and model-driven DL-based solutions are considered for both infinite-precision and coarsely-quantized systems. This concluding chapter summarises the research studies proposed in this thesis. In Section 7.1, the main objective of each chapter and the achieved results are reviewed. Then the potential improvements and future work of the research are discussed in Section 7.2.

7.1 Conclusions

Over the past several years, DL has shown great potential to revolutionize wireless communication systems from the upper layers to the physical layer (PHY). Most of the initial attempts in the literature directly introduce pure data-driven neural networks into wireless signal processing according to the universal approximation theorem. However, these model-agnostic solutions are not very efficient in terms of complexity and convergence, especially in PHY. The main focus of this thesis is to develop efficient receivers with competitive performance, low complexity, and fast convergence via the complementation of expert knowledge and DL techniques. In Chapter 3, we propose a DL-based channel estimator and a signal detector in a model-aided manner, i.e. initializing the DNNs with the aid of conventional algorithms and refining the coarse

estimates by the proposed networks. For a better trade-off between performance and complexity under large-scale MIMO-OFDM systems, Chapter 4 presents a model-based architecture and optimizes it with DL-based techniques, which is called model-driven deep learning. The challenges of channel estimation and quantized detection for massive MU-MIMO-OFDM systems are also addressed by the novel schemes proposed in Chapters 5 and 6. In this section, we will summarize the key findings and contributions of each chapter.

7.1.1 Chapter 3 Conclusion

In Chapter 3, we have proposed RecNet, a model-aided receiver network that realizes channel estimation and signal detection in OFDM systems. The CE-NN, a DL-based channel estimator with lattice-type pilots, can obtain accurate CSI that can be further utilized in signal detection modules. By exploiting the underlying structural features of doubly-selective channels, the proposed CE-NN has strong robustness to the small number of pilots, thus resulting in better spectrum efficiency. Together with the flexibility of our pilot pattern, CE-NN can adapt to different PHY standards like 802.11, LTE, and 5G NR.

Our initial results in [10] demonstrate the superior performance against traditional MMSE-based receivers and state-of-the-art DL-based solutions. To better adjust to the higher-order modulation schemes and further reduce the complexity of RecNet, a series of data preprocessing and model optimization strategies are designed based on the experiments and data analysis. Moreover, we develop two different model configurations of the detection subnet SD-NN to deal with varying degrees of non-linear effects. As a result, our simulation reveals the superiority of RecNet in resisting the influence of non-linear distortions like short CP.

Furthermore, with the enhancement of expert knowledge, both of the two deep subnetworks in the RecNet have the ability for fast convergence within a very small number of training epochs. This ability leads to short training time and a small size of the required dataset. Therefore, the proposed RecNet can efficiently adapt to the dramatic changes of the channel and even different scenarios through fast and affordable retraining. Finally, the low complexity of RecNet has been demonstrated by our complexity comparison. In summary, these features of RecNet are helpful for the deployment of DL-based schemes in realistic communication systems. The idea of model-aided designs in Chapter 3 provides deeper insights into the efficient implementation of deep learning models in wireless signal processing.

7.1.2 Chapters 4 and 5 Conclusions

In Chapters 4 and 5, to ensure reliable communication on realistic channels with serious correlation and frequency selectivity, we propose novel DL-based schemes for signal detection and channel estimation for the case when massive MU-MIMO is combined with OFDM. Different from the SISO-OFDM case in Chapter 3 or small-scale MIMO systems in the literature of DL-based receivers, massive MIMO-OFDM systems have very high-dimensional data to be detected. In such scenarios, millions of learnable parameters lead to substantial computational costs, which hinder the implementation of generic or model-aided DNNs like RecNet. Motivated by these findings, our focus shifts to model-driven deep learning, which utilizes DL tools to improve model-based architectures.

Although DL-based schemes have shown promising performance in MIMO detection, there are still many open issues in this emerging research area, including their complexity, robustness, and integration into full communication systems [67]. In Chapter 4, we have proposed the MMO-Net, a deep unfolding-based detection network for MU-MIMO-OFDM systems. This solution fuses the interpretable ADMM

architecture with over-relaxed design, DU techniques, the differentiable multilevel projection function, and additional learnable parameters. First, compared with existing DL-based MIMO detectors, MMO-Net has relatively low complexity and a small number of trainable parameters. By unfolding ADMM iterations into network layers and learning to optimize step sizes and relaxation parameters jointly, MMO-Net can converge within ten layers. Moreover, for performance and robustness, our simulation results demonstrate the superiority of MMO-Net to the MMSE, CG and several state-of-the-art DL-based MIMO detectors, especially under realistic frequency-selective channels and the full load case of Rx antennas. Finally, as shown in Fig. 4.1, MMO-Net is designed to integrate into the complete MU-MIMO-OFDM systems for 5G and beyond. The efficient implementation scheme proposed in Subsection 4.5.4 can also help MMO-Net achieve a good trade-off between performance and complexity for various requirements.

For high-loaded systems with large numbers of users in Chapter 5, we develop a frequency-orthogonal pilot scheme to save time resource used for pilot transmission. By the special subtractive residual layer, the proposed D-CNN can efficiently denoise the rough CIR in the delay domain. To further enhance estimation performance, a residual RSF-CNN exploits spatial-frequency correlations of channels to refine the output of D-CNN. With the help of the specially designed FFT layer, D-CNN and RSF-CNN can be jointly trained across different domains, leading to the DRSF-CNN which shows advanced performance and fast convergence with low complexity. Furthermore, the proposed detector MMO-Net in Chapter 4 and DRSF-CNN are jointly evaluated as an entire receiver in Subsection 5.5.5, which is one of the first works in the literature.

7.1.3 Chapter 6 Conclusion

In Chapter 6, the challenges of data detection with low-resolution ADCs are discussed in the massive MU-MIMO-OFDM system with high-order modulations and frequency-selective channels, which is the practically more relevant case of future wireless communications. Whereas prior works mainly focus on either quantized MIMO detection with frequency-flat channels or SISO-OFDM detection using model-agnostic DNNs, which perform well against non-linear effects but have a huge number of parameters and high computational complexity.

To efficiently detect the received signals with severe non-linear distortions under the harsh system in this chapter, we have proposed the QMMO-Net, a model-driven detection network. This scheme combines a novel $\boldsymbol{x}$-update with additive gradient correction, a special-designed network skeleton, the differentiable and flexible proximal operator, and state-of-the-art DL techniques. In terms of performance and robustness, our simulation results demonstrate the superiority of QMMO-Net to the conventional MMSE, ADMM, and DL-based MIMO detectors in coarsely quantized massive MIMO-OFDM systems, especially under the cases of high user load.

7.2 Future Work

Although the model-aided and model-driven DL-based schemes proposed in this thesis have shown promising results in terms of performance, complexity, and robustness under massive MIMO-OFDM systems, there are still some aspects of the research that need to be further improved. In this section, we suggest several potential research directions for future work.

7.2.1 Model-aided Deep Learning for Channel estimation and Signal Detection

In Chapter 3, we mainly focus on improving the generic NN-based architectures like fully-connected DNN and CNN in OFDM systems with the aid of expert knowledge. First, this work can be extended to MIMO-OFDM communications. More nonlinear effects like hardware impairment could be considered, and corresponding experiments should also be conducted. Considering the increasing complexity and training overhead under MIMO-OFDM systems, some novel machine learning techniques like ensemble learning and online learning-based fine tuning can be introduced to improve the adaptation and avoid frequent retraining for brand-new scenarios. Moreover, due to the relatively small size of RecNet, the joint optimization between the RecNet algorithm and reconfigurable hardware such as Field Programmable Gate Array (FPGA) is feasible to provide further improvement in energy efficiency.

7.2.2 DL-based Channel Estimation and Signal Detection for Massive MU-MIMO-OFDM Systems

In Chapter 4, a series of novel techniques have been proposed to improve the MMO-Net in terms of performance, complexity, and convergence. Since MMO-Net has superior robustness to the high user load of MU-MIMO-OFDM systems, it is capable of generalizing good performance for varying numbers of users with different degrees of CCI. However, for various modulation schemes, MMO-Net requires different projection functions like existing DU-based detectors. Thus, the self-adaptation to different order modulations should be considered in the future. A possible idea is to treat modulation schemes and corresponding projection functions as a hyper-

parameter, utilize a neural network to learn it, and select the proper projection function based on the received data. Furthermore, efficient training strategies, such as the layer-by-layer way in [57], and fine-tuning for MMO-Net could also be considered as potential improvements.

Compared with the CE-NN in Chapter 3, the DRSF-CNN proposed in Chapter 5 has successfully extended the channel estimation task from simple OFDM systems to massive MU-MIMO-OFDM systems with realistic channels. The superiority of DRSF-CNN is achieved by exploiting channel correlations across different domains. It is worthwhile to further investigate the impact of training channel data on DL-based channel estimators, which is still lacking. For instance, the proper amount of training samples for different channel models is different. Realistic channel models like 3GPP-3D channels in Chapters 4 and 5 are much more complex than i.i.d Gaussian channels, which means more training data is required to learn the underlying features. However, unlike the computer vision field where there are sufficient accessible real-world data sets, the open-access data sets for wireless communications are limited by data regulations and are still under developed [103]. At this stage, we suggest an alternative solution, i.e., enhancing or expanding limited training data via data augmentation techniques like generative adversarial networks. Moreover, the training dataset with mixed SNRs and corresponding training methods may help to improve the robustness of DRSF-CNN to different SNRs.

7.2.3 Deep Unfolding-based Detection for Quantized MIMO-OFDM Systems

Due to their ability to handle non-linear distortions, most existing learning-based detectors for low-precision quantization rely on generic DNN models. To the best of our knowledge, the research work in Chapter 6 is one of the first attempts at fusing model-based architecture and deep unfolding for quantized massive MIMO-OFDM detection. In practice, the CSI used for detection is estimated via pilot symbols rather than perfectly known. Therefore, an efficient channel estimation scheme is important for coarsely quantized MIMO-OFDM receivers, which could be developed together with the proposed QMMO-Net in the future. Correspondingly, more comprehensive experiments with channel estimation errors should also be conducted to provide more insights into DL-based MU-MIMO-OFDM detection tasks with low-resolution ADCs.

Bibliography

[1] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Cisco visual networking index (vni) complete forecast update, 2017–2022," *Americas/EMEAR Cisco Knowledge Network (CKN) Presentation*, pp. 1–30, 2018.

[2] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Communications*, vol. 14, no. 11, pp. 92–111, 2017.

[3] D. Wang, Y. Zhang, H. Wei, X. You, X. Gao, and J. Wang, "An overview of transmission theory and techniques of large-scale antenna systems for 5G wireless communications," *Science China Information Sciences*, vol. 59, no. 8, pp. 1–18, 2016.

[4] X. You, C. Zhang, X. Tan, S. Jin, and H. Wu, "Ai for 5g: research directions and paradigms," *Science China Information Sciences*, vol. 62, no. 2, pp. 1–13, 2019.

[5] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[6] S. Payami and F. Tufvesson, "Channel measurements and analysis for very large array systems at 2.6 ghz," in *2012 6th European Conference on Antennas and Propagation (EUCAP)*, 2012, pp. 433–437.

[7] C.-X. Wang, S. Wu, L. Bai, X. You, J. Wang *et al.*, "Recent advances and future challenges for massive mimo channel measurements and models," *Science China Information Sciences*, vol. 59, no. 2, pp. 1–16, 2016.

[8] S. Ali, W. Saad, N. Rajatheva, K. Chang, D. Steinbach, B. Sliwa, C. Wietfeld, K. Mei, H. Shiri, H.-J. Zepernick *et al.*, "6g white paper on machine learning in wireless communication networks," *arXiv preprint arXiv:2004.13875*, 2020.

[9] T. O'shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[10] C. Liu and T. Arslan, "Recnet: Deep learning-based OFDM receiver with semi-blind channel estimation," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–4.

[11] C. Liu, J. Thompson, and T. Arslan, "Ofdm receivers with semi-blind channel estimation based on deep neural networks," *IEEE Transactions on Cognitive Communications and Networking*, 2022. (Under Review).

[12] C. Liu, J. Thompson, and T. Arslan, "A deep unfolding network for massive multi-user mimo-ofdm detection," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 2405–2410.

[13] C. Liu, J. Thompson, and T. Arslan, "Deep learning based receivers for massive mimo ofdm systems with high user load," *IEEE Transactions on Wireless Communications*, 2022. (Under Review).

[14] C. Liu, J. Thompson, and T. Arslan, "Deep unfolding-based detection for quantized massive mu-mimo-ofdm systems," in *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*. IEEE, 2022, pp. 1–5.

[15] T. S. Rappaport *et al.*, *Wireless communications: principles and practice 2/E*. Prentice Hall, 2001.

[16] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM wireless communications with MATLAB*. John Wiley & Sons, 2010.

[17] H. T. Friis, "A note on a simple transmission formula," *Proceedings of the IRE*, vol. 34, no. 5, pp. 254–256, 1946.

[18] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[19] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, "An overview of MIMO communications-a key to gigabit wireless," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 198–218, 2004.

[20] L. Cimini, "Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing," *IEEE transactions on communications*, vol. 33, no. 7, pp. 665–675, 1985.

[21] L. Zhaogan, R. Yuan, Z. Taiyi, and W. Liejun, "Multiuser mimo ofdm based tdd/tdma for next generation wireless communication systems," *Wireless Personal Communications*, vol. 52, no. 2, pp. 289–324, 2010.

[22] K. P. Kongara, P. J. Smith, and L. M. Garth, "Frequency domain variation of eigenvalues in adaptive mimo ofdm systems," *IEEE transactions on wireless communications*, vol. 10, no. 11, pp. 3656–3665, 2011.

[23] S. Ajey, B. Srivalli, and G. Rangaraj, "On performance of mimo-ofdm based lte systems," in *2010 International Conference on Wireless Communication and Sensor Computing (ICWCSC)*, 2010, pp. 1–5.

[24] A. Zaidi, F. Athley, J. Medbo, U. Gustavsson, G. Durisi, and X. Chen, *5G Physical Layer: Principles, Models and Technology Components*. USA: Academic Press, Inc., 2018.

[25] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, 2010.

[26] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive mimo for next generation wireless systems," *IEEE communications magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[27] Y. d. J. Bultitude and T. Rautiainen, "Ist-4-027756 winner ii d1. 1.2 v1. 2 winner ii channel models," *EBITG, TUI, UOULU, CU/CRC, NOKIA, Tech. Rep*, 2007.

[28] A. Osseiran, J. F. Monserrat, and P. Marsch, *5G Mobile and Wireless Communications Technology*, 1st ed. USA: Cambridge University Press, 2016.

[29] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "Quadriga: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 6, pp. 3242–3256, 2014.

[30] T. L. Marzetta, "How much training is required for multiuser mimo?" in *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, 2006, pp. 359–363.

[31] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO detection techniques: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3109–3132, 2019.

[32] W.-K. Ma, T. N. Davidson, K. M. Wong, Z.-Q. Luo, and P.-C. Ching, "Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous cdma," *IEEE transactions on signal processing*, vol. 50, no. 4, pp. 912–922, 2002.

[33] Y. Hu, Z. Wang, X. Gaol, and J. Ning, "Low-complexity signal detection using cg method for uplink large-scale mimo systems," in *2014 IEEE international conference on communication systems*. IEEE, 2014, pp. 477–481.

[34] Y. Xue, C. Zhang, S. Zhang, and X. You, "A fast-convergent pre-conditioned conjugate gradient detection for massive mimo uplink," in *2016 IEEE International Conference on Digital Signal Processing (DSP)*. IEEE, 2016, pp. 331–335.

[35] M. Chaudhary, N. K. Meena, and R. S. Kshetrimayum, "Local search based near optimal low complexity detection for large mimo system," in *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2016, pp. 1–5.

[36] O. Castaneda, T. Goldstein, and C. Studer, "Data detection in large multi-antenna wireless systems via approximate semidefinite relaxation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2334–2346, 2016.

[37] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[38] Z.-H. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks." in *IJCAI*, 2017, pp. 3553–3559.

[39] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[40] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[41] ITU, *ITU-T Y.3170-series – Machine learning in future networks including IMT-2020: Use cases (Study Group 13)*, Std., 2019.

[42] M. S. Parwez, D. B. Rawat, and M. Garuba, "Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2058–2065, 2017.

[43] L.-C. Wang and S.-H. Cheng, "Data-driven resource management for ultra-dense small cells: An affinity propagation clustering approach," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 3, pp. 267–279, 2018.

[44] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[45] W. Xie, S. Hu, C. Yu, P. Zhu, X. Peng, and J. Ouyang, "Deep learning in digital modulation recognition using high order cumulants," *IEEE Access*, vol. 7, pp. 63 760–63 766, 2019.

[46] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 2326–2330.

[47] S. Han, Y. Oh, and C. Song, "A deep learning based channel estimation scheme for ieee 802.11 p systems," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[48] Y. Yang, F. Gao, X. Ma, and S. Zhang, "Deep learning-based channel estimation for doubly selective fading channels," *IEEE Access*, vol. 7, pp. 36 579–36 589, 2019.

[49] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmwave massive mimo systems," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 852–855, 2018.

[50] J. Gao, C. Zhong, G. Y. Li, and Z. Zhang, "Deep learning based channel estimation for massive mimo with hybrid transceivers," *IEEE Transactions on Wireless Communications*, 2021.

[51] E. Balevi, A. Doshi, and J. G. Andrews, "Massive mimo channel estimation with an untrained deep neural network," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2079–2090, 2020.

[52] S. Dörner, S. Cammerer, J. Hoydis, and S. Ten Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2017.

[53] N. Farsad and A. Goldsmith, "Neural network detection of data sequences in communication systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5663–5678, 2018.

[54] X. Yan, F. Long, J. Wang, N. Fu, W. Ou, and B. Liu, "Signal detection of mimo-ofdm system based on auto-encoder and extreme learning machine," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1602–1606.

[55] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 77–83, 2019.

[56] L. Xu, F. Gao, W. Zhang, and S. Ma, "Model aided deep learning based mimo ofdm receiver with nonlinear power amplifiers," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–6.

[57] Y. Wei, M.-M. Zhao, M. Hong, M.-J. Zhao, and M. Lei, "Learned conjugate gradient descent network for massive MIMO detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6336–6349, 2020.

[58] M.-H. Hsieh and C.-H. Wei, "Channel estimation for ofdm systems based on comb-type pilot arrangement in frequency selective fading channels," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 1, pp. 217–225, 1998.

[59] H. Zarrinkoub, *Understanding LTE with MATLAB: from mathematical modeling to simulation and prototyping*. John Wiley & Sons, 2014.

[60] Y. Li, L. J. Cimini, and N. R. Sollenberger, "Robust channel estimation for ofdm systems with rapid dispersive fading channels," *IEEE Transactions on communications*, vol. 46, no. 7, pp. 902–915, 1998.

[61] C. Rezgui and K. Grayaa, "An enhanced channel estimation technique with adaptive pilot spacing for ofdm system," in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2016, pp. 1–4.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[63] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[64] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based csi feedback approach for time-varying massive mimo channels," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 416–419, 2018.

[65] W. Xu, Z. Wu, Y.-L. Ueng, X. You, and C. Zhang, "Improved polar decoder based on deep learning," in *2017 IEEE International workshop on signal processing systems (SiPS)*. IEEE, 2017, pp. 1–6.

[66] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based mimo communications," *arXiv preprint arXiv:1707.07980*, 2017.

[67] Neev Samuel, Tzvi Diskin, and Ami Wiesel, "Learning to detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.

[68] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[69] X. Gao, S. Jin, C.-K. Wen, and G. Y. Li, "Comnet: Combination of deep learning and expert knowledge in ofdm receivers," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2627–2630, 2018.

[70] E. Balevi and J. G. Andrews, "One-bit ofdm receivers via deep learning," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4326–4336, 2019.

[71] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[72] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[73] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[76] Z. Xu and J. Sun, "Model-driven deep-learning," *National Science Review*, vol. 5, no. 1, pp. 22–24, 2018.

[77] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[78] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40–60, 2013.

[79] S. Wu, L. Kuang, Z. Ni, J. Lu, D. Huang, and Q. Guo, "Low-complexity iterative detection for large-scale multiuser MIMO-OFDM systems using approximate message passing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 902–915, 2014.

[80] H. Prabhu, O. Edfors, J. Rodrigues, L. Liu, and F. Rusek, "Hardware efficient approximative matrix inversion for linear precoding in massive MIMO," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2014, pp. 1700–1703.

[81] L. Dai, X. Gao, X. Su, S. Han, I. Chih-Lin, and Z. Wang, "Low-complexity soft-output signal detection based on Gauss-Seidel method for uplink multiuser large-scale MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4839–4845, 2014.

[82] B. Yin, M. Wu, J. R. Cavallaro, and C. Studer, "Conjugate gradient-based soft-output detection and precoding in massive MIMO systems," in *2014 IEEE Global Communications Conference*. IEEE, 2014, pp. 3696–3701.

[83] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," in *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*. IEEE, 2019, pp. 266–271.

[84] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," 2017. [Online]. Available: https://arxiv.org/abs/1706.01151

[85] M.-W. Un, M. Shao, W.-K. Ma, and P. Ching, "Deep MIMO detection using admm unfolding," in *2019 IEEE Data Science Workshop (DSW)*. IEEE, 2019, pp. 333–337.

[86] V. Corlay, J. J. Boutros, P. Ciblat, and L. Brunel, "Multilevel MIMO detection with deep learning," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 1805–1809.

[87] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2018, pp. 584–588.

[88] Hengtao He, Chao-Kai Wen, Shi Jin and Geoffrey Ye Li, "Model-driven deep learning for MIMO detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1702–1715, 2020.

[89] P. Zheng, Y. Zeng, Z. Liu, and Y. Gong, "Deep learning based trainable approximate message passing for massive MIMO detection," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[90] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5635–5648, 2020.

[91] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 644–658, 2015.

[92] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[93] J. Eckstein, "Parallel alternating direction multiplier decomposition of convex programs," *Journal of Optimization Theory and Applications*, vol. 80, no. 1, pp. 39–62, 1994.

[94] Ö. T. Demir and E. Björnson, "Admm-based one-bit quantized signal detection for massive mimo systems with hardware impairments," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 9120–9124.

[95] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 2017.

[96] H. He, "https://github.com/hehengtao/oamp-net," 2020.

[97] M. Borgmann and H. Bolcskei, "Interpolation-based efficient matrix inversion for MIMO-OFDM receivers," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 2, 2004, pp. 1941–1947 Vol.2.

[98] C. Sun, X. Gao, S. Jin, M. Matthaiou, Z. Ding, and C. Xiao, "Beam division multiple access transmission for massive MIMO communications," *IEEE Transactions on Communications*, vol. 63, no. 6, pp. 2170–2184, 2015.

[99] M. K. Ozdemir, H. Arslan, and E. Arvas, "Toward real-time adaptive low-rank lmmse channel estimation of mimo-ofdm systems," *IEEE transactions on wireless communications*, vol. 5, no. 10, pp. 2675–2678, 2006.

[100] J. Ma and L. Ping, "Data-aided channel estimation in large antenna systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3111–3124, 2014.

[101] C. Qi, Y. Huang, S. Jin, and L. Wu, "Sparse channel estimation based on compressed sensing for massive MIMO systems," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 4558–4563.

[102] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO," *IEEE Transactions on Signal Processing*, vol. 63, no. 23, pp. 6169–6183, 2015.

[103] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 93–99, 2019.

[104] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 2019.

[105] Y. Liao, Y. Hua, X. Dai, H. Yao, and X. Yang, "Chanestnet: A deep learning based channel estimation for high-speed scenarios," in *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–6.

[106] Q. Hu, F. Gao, H. Zhang, S. Jin, and G. Y. Li, "Deep learning for channel estimation: Interpretation, performance, and comparison," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2398–2412, 2020.

[107] Y. Liao, Y. Hua, and Y. Cai, "Deep learning based channel estimation algorithm for fast time-varying mimo-ofdm systems," *IEEE Communications Letters*, vol. 24, no. 3, pp. 572–576, 2019.

[108] P. Dong, H. Zhang, G. Y. Li, I. S. Gaspar, and N. NaderiAlizadeh, "Deep cnn-based channel estimation for mmwave massive mimo systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 989–1000, 2019.

[109] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based channel estimation for massive mimo systems," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1228–1231, 2019.

[110] Chang-Jae Chun, Jae-Mo Kang and I-Min Kim, "Deep learning-based joint pilot design and channel estimation for multiuser mimo channels," *IEEE Communications Letters*, vol. 23, no. 11, pp. 1999–2003, 2019.

[111] C. Studer and G. Durisi, "Quantized massive mu-mimo-ofdm uplink," *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2387–2399, 2016.

[112] R. Negi and J. Cioffi, "Pilot tone selection for channel estimation in a mobile ofdm system," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 3, pp. 1122–1128, 1998.

[113] I. Barhumi, G. Leus, and M. Moonen, "Optimal training design for mimo ofdm systems in mobile wireless channels," *IEEE Transactions on signal processing*, vol. 51, no. 6, pp. 1615–1624, 2003.

[114] H. Minn and N. Al-Dhahir, "Optimal training signals for mimo ofdm channel estimation," *IEEE transactions on wireless communications*, vol. 5, no. 5, pp. 1158–1168, 2006.

[115] D. Chu, "Polyphase codes with good periodic correlation properties (corresp.)," *IEEE Transactions on information theory*, vol. 18, no. 4, pp. 531–532, 1972.

[116] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[117] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE journal of selected topics in signal processing*, vol. 8, no. 5, pp. 742–758, 2014.

[118] R. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, 1999.

[119] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5g be?" *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[120] S.-Y. Lien, S.-L. Shieh, Y. Huang, B. Su, Y.-L. Hsu, and H.-Y. Wei, "5G new radio: Waveform, frame structure, multiple access, and initial access," *IEEE communications magazine*, vol. 55, no. 6, pp. 64–71, 2017.

[121] C. Risi, D. Persson, and E. G. Larsson, "Massive mimo with 1-bit adc," 2014. [Online]. Available: https://arxiv.org/abs/1404.7736

[122] A. Mezghani and J. A. Nossek, "Belief propagation based mimo detection operating on quantized channel output," in *2010 IEEE International Symposium on Information Theory*. IEEE, 2010, pp. 2113–2117.

[123] A. Mezghani, M. Rouatbi, and J. A. Nossek, "An iterative receiver for quantized mimo systems," in *2012 16th IEEE Mediterranean Electrotechnical Conference*. IEEE, 2012, pp. 1049–1052.

[124] S. Wang, Y. Li, and J. Wang, "Multiuser detection in massive spatial modulation mimo with low-resolution adcs," *IEEE Transactions on Wireless Communications*, vol. 14, no. 4, pp. 2156–2168, 2015.

[125] N. Shlezinger, R. Fu, and Y. C. Eldar, "Deepsic: Deep soft interference cancellation for multiuser mimo detection," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1349–1362, 2020.

[126] Z. Lu, L. Wei, and Y. Xu, "An improved one-bit ofdm receiver based on model-driven deep learning," in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE, 2019, pp. 2108–2112.

[127] D. Kim and N. Lee, "Machine learning based detections for mmwave two-hop mimo systems using one-bit transceivers," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[128] A. Klautau, N. González-Prelcic, A. Mezghani, and R. W. Heath, "Detection and channel equalization with deep learning for low resolution mimo systems," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 1836–1840.

[129] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[130] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[131] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[132] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.