Deep Learning Designs for Physical Layer Communications

Abdullahi Mohammad (SN: 17093730)

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy (Ph.D.)

of

University College London.

Department of Electronic and Electrical Engineering

University College London

May 19, 2022

I, Abdullahi Mohammad (SN: 17093730), declare and confirm that the work presented in this thesis is exclusively my own. Where information is obtained from other sources is dully cited and appropriately referenced. I also confirm that all the information I submitted in this thesis is correct, accurate, and valid, and I will present the supporting documents therein when required.

Abstract

Wireless communication systems and their underlying technologies have undergone unprecedented advances over the last two decades to assuage the ever-increasing demands for various applications and new emerging technologies. However, the traditional signal processing schemes and algorithms for wireless communications cannot handle the upsurging complexity associated with fifth generation (5G) and beyond communication systems due to network expansion, emerging technologies, high data-rate, and the ever-increasing demands for low latency.

This thesis extends the traditional downlink transmission schemes to deep-learning based precoding and detection techniques that are hardware-efficient and of lower complexity than the current state-of-the-art. The thesis focuses on: precoding/beamforming in massive multiple-inputs-multiple-outputs (MIMO), signal detection and lightweight neural network (NN) architectures for precoder and decoder designs. We introduce a learning-based precoder design via constructive interference (CI) that performs the precoding on a symbol-by-symbol basis. Instead of conventionally training a NN without considering the specifics of the optimisation objective, we unfold a power minimisation symbol level precoding (SLP) formulation based on the interior-point-method (IPM) proximal 'log' barrier function. Furthermore, we propose a concept of NN compression, where the weights are quantised to lower numerical precision formats based on binary and ternary quantisations. We further introduce a stochastic quantisation technique, where parts of the NN weight matrix are quantised while the remaining is not. Finally, we propose a systematic complexity scaling of deep neural network (DNN) based MIMO detectors. The model uses a fraction of the DNN inputs by scaling their values through weights that follow monotonically non-increasing functions. Furthermore, we investigate performance complexity tradeoffs via regularisation constraints on the layer weights such that, at inference, parts of network layers can be removed with minimal impact on the detection accuracy.

Simulation results show that our proposed learning based techniques offer better complexity-vs-BER (bit-error rate) and complexity-vs-transmit power performances compared to the state-of-the-art MIMO detection and precoding techniques.

Impact Statement

One of the essential enabling technologies for the fifth generation (5G) and beyond wireless communications is massive multiple-inputs-multiple-outputs (m-MIMO), where the base station (BS) is endowed with hundreds or thousands of antennas. While 5G technology has many benefits, such as high data rate, ultra-low latency, improved spectral efficiency, and increased connectivity, the hardware requirements and circuit power consumption grow proportionally with the number of BS antennas due to infrastructural costs. The learning-based techniques proposed in this thesis present new methods for the cost-efficient deployment of m-MIMO compared to the current systems with traditional excessive power-demanding BS antenna circuits.

Furthermore, as the 5G network is being rolled out in different parts of the world, it is expected to accommodate over 50 billion connected devices generating massive data and is envisaged to grow by 12% annually. Similarly, with the advent of 5G, mobile users alone are envisioned to reach 7 billion by 2023. This exponential growth of connected terminals at the wireless network edge results in the surging complexity of the network, which is challenging for the current signal processing schemes, such as precoding and equalisation, to handle due to the computational costs involved in their implementation on practical wireless communication systems. Moreover, CO_2 is the primary source of greenhouse gas responsible for the greenhouse effect, and the new capabilities brought by the 5G network potentially pose an environmental threat that increases CO_2 emissions. An optimal precoding design is needed to reduce the power required for efficient downlink transmission in a multiuser multiple-inputs-single-output (MISO) system in order to ameliorate the CO_2 emissions from the BS. The proposed machine learning (ML) solutions presented in this research will provide a new paradigm shift for making advanced precoding practical, reducing complexity, 5G infrastructure, bridging the digital divide, reducing transmit power, helping with climate change and busting the revenue generation of the mobile providers. Beyond providing learning-based signal processing capabilities for physical layer communications, this research opens up a new approach for scalable learning framework designs that will facilitate fast learning on the embedded systems that are resource constrained, thus easing the deployment of trained models on the device edge. Overall, the findings presented in this research work provide intuition behind learning-based solutions that can be potentially explored to aid the actualisation of more robust 5G networks and even beyond.

Acknowledgements

ALHAMDULILLAH!! All praises are due to Allah, the Lord of all the worlds, for the gift of health and life. Blessings and beautiful salutations are upon our beloved Prophet Muhammad (SAW), his households, guided companions and all those that follow them in righteous deed till the day of recompense.

Firstly, I would like to register my profound gratitude and appreciation to my principal supervisor Prof. Christos Masouros and my second supervisor Prof. Yiannis Andreopoulos for their patience, consistent motivation, constructive criticisms, immense support, and guidance during four years of my PhD training. I want to express my gratitude to Prof. Izzat Darwazeh for making it possible to complete this programme. I would also like to uniquely thank my beloved mother for her unflinching love, support and constant prayers. Mother, you always keep my spirit and hope high. To my better half, Maryam and beloved children, Sumayyah, Sa'adah and bay Abdallah, who was born amidst the Covid19 pandemic and the pressure of pursuing a PhD, I say big thank you for your patience and for always being there for me through these challenging times.

My special thanks also go to Hajiya Rabi of the Petroleum Technology Development Fund (PTDF) and the Federal Republic of Nigeria for providing this rare opportunity to study PhD in one of the best world's leading universities, the Global University (University College London). My sincere appreciation goes to my second family in London, Sheikh Osman Nur Mohammed, his beloved wife, Rahma Hassan Rabileh and their children, for the love, support, prayers and encouragement. Abbah, I will forever remain indebted to you. My particular appreciation goes to my brothers and true friends, Dr Yusuf Abubakar Sha'aban and Dr Abubkar

Sha'aban, for your constant prayers, encouragement and inspiration. My special thanks go to my mentor, Prof Muhammad Bashir Mu'azu, for the mentor-ship and moral supports throughout my academic journey.

Furthermore, I would like to use this opportunity to appreciate my fellow research colleague and a friend, Dr Mahmoud Tukur Kabir, who was there whenever I needed support and advice; thank you very much. To my other research colleagues, Dr Abdelhamid Salem, Dr Temitope Odedeyi, Dr Zhongxiang Wei, Dr. Shakir Badmos (Baba Labiba) and Sheikh Motasem Alsawadi, for their support and encouragement during the entire period of my study. My sincere appreciation goes to my siblings, all members of my extended family and friends for their good wishes and prayers. I say a big thank you to Malam Umar Sada Zaria and his family for the prayers and moral support during my stay in London. I cannot end this without thanking my uncle and in-law, Alhaji Abdullahi M. Barau and his brother, Alhaji Abubakar for their genuine love and moral supports. Baba, your prayers and constant support towards my family while I was away for this study are highly appreciated, and I pray to Allah to reward you in a way that you have never imagined. Finally, to my special friends: Arch. Shehu Dikko, Engr Saidu Muhammad Waziri, Mahmoud Nuhu Babajo, Engr Umar Musa Adam, Dr M. Dikko Al-Mustapha, Ismail Dari, a brother and a friend, Bello Ahmed Wakili, uncles and friends, Husaini Umar and Abubakar Umar, I am indebted to you all.

Contents

List of Abbreviations

5G	Fifth Generation
AI	Artificial Intelligence
AMP	Approximate Message Passing
ANN	Artificial Neural Networks
APB	Auxiliary Processing Block
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BLP	Block Level Precoding
BPSK	Binary Phase-Shift Keying
BS	Base Station
CI	Constructive Interference
CNN	Convolutional Neural Network
CS	Compressed Sensing
DL	Deep Learning
DNN	Deep Neural Network
DPC	Dirty Paper Coding

FDD	Frequency Division Duplex
FIR	Finite Impulse Response
IoT	Internet of Things
IPM	Interior Point Method
ISI	Inter-Symbol Interference
LE	Linear Estimation
LoS	Line-of-Sight
m-MIMO	massive MIMO
MACs	Multiply Accumulates
MAP-D	Maximum a Posteriori Detector
MF-P	Match Filter Precoder
MFD	Match Filter Detector
MIMO	Multiple-Inputs-Multiple-Outputs
MISO	Multiple-Inputs-Single-Output
ML	Machine Learning
ML-D	Maximum Likelihood Detector
MLP	Multilayer Perceptron
MMSE	Minimum Mean Square-Error
MMSE-D	Minimum Mean-Square Error Detector
MRC	Maximum Ratio Combining
MRT	Maximum Ratio Transmission

MSE	Mean-Squared Error
MU-MISO	Multi-User Multiple-Inputs-Single-Output
MUI	Multi-User-Interference
NLE	Nonlinear Estimation
NN	Neural Network
OAMP	Orthogonal AMP
OAMP-Net	orthogonal Approximate Message Passing Deep Network
PBFs	Proximity Barrier Functions
PCA	Principal Component Analysis
PSK	Phase-Shift Keying
PUM	Parameter Update Module
QAM	Quadrature Amplitude Modulation
QCOP	Quadratically Constrained Quadratic Programming
QoS	Quality of Service
QPSK	Quadrature Phase-Shift Keying
RZF-P	Regularised Zero-Forcing Precoder
SD	Sphere Decoding
SDP	Semidefinite Programming
SDR	Semidefinite Relaxation
SER	Symbol-Error-Rate
SINR	Signal-to-Interference-Noise Ratio

SLP	Symbol Level Precoding
SLP-DNet	Symbol Level Precoding Deep Neural Network
SNR	Signal-to-Noise Ratio
SOC	Second Order Cone
SQ	Stochastic quantisation
STE	Straight-through Estimator
SVM	Support Vector Machine
TDD	Time Division Duplex
THP	Tomlinson-Harashima Precoding
TPG-Net	Trainable Projected Gradient Detector
UE	User Equipment
V-BLAST	Vertical-Bell Laboratories Layered Space-Time
VP	Vector Perturbation
WeSNet	Weight-Scaling Neural-Network
WMMSE	Weighted Minimum Mean-Squared Error
WSR	Weighted Sum Rate
ZF-D	Zero-Forcing Detector
ZF-P	Zero-Forcing Precoder

List of Symbols

a	Scalar
$\mathbf{a}$	Vector
$\mathbf{A}$	Matrix
j	Imaginary unit
$\mathbb{C}^{m\times n}$	A $m \times n$ matrix in the complex set
$\mathbb{R}^{m\times n}$	A $m \times n$ matrix in the real set
$\mathbf{A} \succeq 0$	Matrix $\mathbf{A}$ is positive semi-definite
$\mathcal{N}(\alpha,\beta)$	Complex normal distribution
$\sim\mathcal{U}(\alpha,\beta)$	Real uniform distribution
$\Re(\cdot)$	Real part of a complex scalar, vector or matrix
$\Im(\cdot)$	Imaginary part of a complex scalar, vector or matrix
$(\cdot)^T$	Transpose
$(\cdot)^H$	Conjugate transpose or Hermitian transpose
$\mathrm{tr}\{\cdot\}$	Trace of a matrix
$(\cdot)^{-1}$	Inverse of a square matrix
$(\cdot)^{\dagger}$	Moore-Penrose inverse
$\lvert\cdot\rvert$	Absolute value or modulus
$\lVert\cdot\rVert_1$	l_1-norm
$\lVert\cdot\rVert_2$	l_2-norm
$\mathscr{O}(\cdot)$	Order of numerical operations
$\mathcal{L}(\cdot)$ and $\mathscr{L}(\cdot)$	Loss functions
$\Psi\{\cdot\}$	Set of learnable parameters
$\mathcal{J}(\cdot)$	Jacobian matrix

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Recent studies have shown that there has been an exponential growth in wireless devices and data traffic over the last few decades, which is envisaged to increase by more than 50% annually in the next couple of years [1]. Consequently, the increasing need for higher data rates has spurred both academia and industry to come up with new techniques. Among the techniques, MIMO has been widely recognised as the most promising for future wireless communication systems [2]. Most existing research work on the MIMO systems has been carried out assuming perfect hardware settings while neglecting imperfections in the hardware elements. The large-scale antenna system known as massive MIMO (m-MIMO) proposed in [3–5] has been proven to be one of the most promising techniques for 5G wireless communications. Theoretically, m-MIMO is similar to the traditional multiple-antenna system, having many antennas extended to the order of hundreds or thousands [5]. Unlike traditional MIMO, m-MIMO systems offer unparalleled benefits, such as momentous high throughput and ultra-low bit error rates [4]. While the benefits from m-MIMO systems are intriguing, they are power-inefficient and have excessive hardware demands [6]. Equivalently, this requirement dramatically raises the hardware budget and, more notably, the resulting power consumption for m-MIMO at the base station (BS). To address this pressing issue, signal processing strategies that are both cost-effective and power-inefficient are needed for reliable downlink signal transmission to balance performance and hardware complexity.

Correspondingly, 5G and beyond wireless network technologies are expected to support enhanced broadband, massive machine-type communications, low latency and ultra-reliable communications [7]. This means that future wireless communication technologies must deal with a large amount of wireless data and adapt to challenging radio environments while satisfying the user's high data rate and speed requirements. The implementation of 5G and beyond is based on the evolution of the existing wireless technologies orchestrated by other new radio schemes to mollify the challenges and the requirements that the current radio network cannot support [8]. These new radio schemes include m-MIMO systems, device-to-device, ultra-reliable, Internet of Things (IoT), and massive machine-enabled communications [9]. Together, they provide a framework for 5G to accommodate an increase in mobile data volume while extending to other application domains that mobile communications can support beyond 2030 [10]. This requires that some intelligence be integrated into the future wireless communication systems to actively adapt to changes in the environmental settings while satisfying the requirements for high speed, quality of service (QoS), spectrum efficiency and low latency [11, 12]. The machine learning (ML) is a popular class of artificial intelligence (AI) that has the potential to deal with high-volume of data and challenges of emerging technologies for future wireless communications.

Thus far, there is hardly any field where ML algorithms, and more specifically deep-learning (DL) has not been tried for various tasks. The breakthroughs are evident in many fields, such as computer vision, natural language processing, machine translation, human-computer interaction, etc [1]. The successful application of DL in these fields is due to the availability of massive labelled data and sufficient computational resources, which may not be applicable in some areas. However, finding the required network learning architecture is still a bottleneck, mainly due to the lack of a solid theoretical basis between the network topology and performance [10, 13]. To this day, the network topology is still an engineering practice rather than scientific research, acknowledging the fact that most existing DL approaches lack theoretical foundations. The typical limitations of the DL approach are the difficulties in network design, interpretability and a lack of understanding of its generalisation ability. These factors may hinder the standardisation of DL and its commercialisation across different domains, such as wireless communications. For instance, there is already a solid theoretical foundation in wireless communications that has led to capacity-achieving algorithms for models that have been shown to work well in practice. In addition, the problems are typically signal-to-noise ratio (SNR) based and may not require very advanced loss functions. In contrast, computer vision and other domains do not have such theoretical foundations because, for example, it is known that SNR does not correlate well with human vision or language perception in the brain. The key factors that motivate the adoption of ML in wireless communications are summarised below:

- **Network Complexity:** The increasing complexity of wireless networks due to emerging applications, topologies, dynamic scenarios etc., renders applying classical signal processing techniques mathematically intractable. However, ML can intelligently leverage the available data associated with future wireless communication to learn complex physical layer wireless communication systems. ML solutions can reduce network complexity while providing effective and promising performance.

- **Algorithm Inefficiency:** Many algorithms designed for cellular networks have proven to show either sub-optimal or optimal performance and they are computationally expensive to implement in a practical sense. On many occasions, engineers are left with no option but to use a heuristic approach based on simple decision principles to design a communication system. For example, the complexity of building a MIMO receiver with N_t spatial antennas and $\mathbb{M}$ constellation using maximum likelihood is of the order of $\mathscr{O}(\mathbb{M}^{N_t})$. Similarly, linear and optimisation-based receivers are sub-optimal, some with prohibitive polynomial complexities. On the other hand, an ML-based decoder can be designed to provide both low complexity and competitive performance.

Given the indispensability of multiuser interference (MUI) management via various precoding techniques at the BS and signal detection at the receiver sides, this thesis focuses on designing novel learning-based and memory-efficient frameworks and transmission strategies for multiple-antenna systems.

1.2 Aim and Motivation

While signal processing methods at the transmitter and receiver sides (BS and mobile end user), such as precoding and signal detection strategies, have been widely researched for multiple-antenna systems, many unsolved areas still require further investigation, particularly for large-scale MIMO systems. It is well-known that traditional nonlinear optimisation-based precoding via constructive interference [14–16] and signal detection [17–20] techniques have achieved considerable performance gain over their linear counterparts. However, implementing such signal processing techniques on practical m-MIMO systems is exceptionally challenging, albeit their impressive performances due to the size of the problems and the nonlinearity of the physical components involved, resulting in prohibitive computational complexity.

For multiuser downlink precoding, recently, closed-form low-complexity precoding designs based on the concept of constructive interference have been proposed in [21]. While the authors have shown that their proposed approaches can achieve a more favourable performance-complexity tradeoff, it is unclear if such techniques can handle the enormous amount of data associated with modern and future wireless communication systems. The ability of ML algorithm to handle massive data has recently sparked up research interests in using ML-based precoding schemes. Following this development, several research papers using ML for precoding [12, 22, 23] and signal detection [24–26] have been written over the last few years.

Similar to the approach in computer vision, the existing ML schemes used in wireless communication are mostly data dependant (data-driven model) without necessarily relying on the mathematical model and expert knowledge [27]. The drawback of this approach is that training such models takes time and the enormous labelled training dataset required, which is usually unavailable in the wireless communication domain. Therefore, the lack of available standard dataset to train a ML model for end-to-end MIMO communications has triggered some research interest for model-driven DNN based on the expert domain knowledge [11, 23]. This approach paves the way for a new paradigm shift from data-driven learning in problems that cannot be expressed by tractable mathematical models or are challenged by algorithmic complexity to a semi-model-driven approach (DL model-driven) based on expert's knowledge.

With relatively low inference complexity, DL-based precoding designs have recently been proposed for multiuser MIMO (MU-MIMO) downlink transmission [24, 28–30]. However, learning-based strategies for wireless physical layer designs use data-driven DL model as a function approximator in a supervised learning mode, which requires labelled training data. This labelled training data is obtained from the analytical solution of the optimisation problem, whose accuracy is bounded by the optimisation algorithm. Alternatively, when possible to have an analytic optimisation objective, the analytical expression is highly non-convex and very high dimensional, such that conventional numerical optimisation is computationally unfeasible. Commensurate with the above, this motivates the development of novel scalable learning-based frameworks for advanced precoding and detection techniques to achieve a more promising performance-complexity tradeoff.

1.3 Main Contributions

The main objective of this thesis is to exploit the potential applications of ML methods in wireless communications. Specifically, to develop efficient and low complexity DL frameworks for wireless physical layer communications. We mainly focus on hardware efficient learning-based solutions for MIMO signal equalisation/detector and precoding designs. Our contributions are summarised below:

- Design an unsupervised DNN-based SLP framework termed SLP-DNet (Chapter 4) for multiuser downlink transmission. To the best of our knowledge, this is the first work that tries to derive an unsupervised deep learning framework by unfolding a constrained power minimisation problem based on SLP. The unique feature of SLP-DNet is that it is built by exploiting the domain knowledge and translating it into a learning-based model via a proximal interior point method (IPM) approach. The SLP-DNet uses the original objective function as a loss function by learning the associated Lagrange multipliers with additional an l_2-norm regularisation term to aid training convergence.

- Introduce the concept of NN compression via weight quantisation to reduce the size of the DNN-based SLP network (Chapter 5). The proposed architectures focus on a DNN with realistic finite precision weights and adopt an unsupervised DL based SLP model (SLP-DNet). We than introduce a stochastic quantisation (SQ) technique to obtain corresponding quantised versions of the full-precision SLP-DNet called SLP-SQDNet, where parts or all the entries of the weight matrix are quantised.
- Propose an NN based MIMO detector where we introduce the concept of monotonic non-increasing profile function to scale each layer of the NN in order to allow the network to dynamically learn the best attenuation strategy for its own weights during training for low complexity MIMO detector design in Chapter 6.

1.4 Thesis Organisation

After this introductory chapter, the thesis is structured as described below:

Chapter 2 reviews the fundamental concepts relevant to 5G and beyond communication systems. Specifically, this chapter presents a synopsis of different precoding and MIMO detection techniques for efficient transmitter and receiver designs. On the transmitter side or BS, emphasis on the precoding designs through interference exploitation on a symbol-by-symbol-level basis. On the other hand, several MIMO detection techniques are reviewed for efficient signal recovery at the receiver.

Chapter 3 gives the general theoretical background of ML and its applications in different domains, focusing only on those used in this thesis. The chapter introduces the conceptual framework of ML, particularly DL, for designing signal processing strategies for physical layer communications. The benefit of using DL for end-to-end wireless communication systems design is underlined. The chapter also describes how data-driven and model-driven DL methods can be joined together to improve the performance of the learning-based signal processing methods. Similarly, an overview of the most popular learning-based MIMO detectors and learning-based precoders are also presented.

Chapter 4 proposes unsupervised learning-based precoding schemes for a multiuser downlink multiple-inputs-single-output (MISO) system. The proposed learning system exploits the CI for the power minimisation problem subject to given QoS constraints. A domain knowledge is used to design unsupervised learning architectures by unfolding the IPM barrier *'log'* function based on the power minimisation formulation. The proposal is extended to robust precoding designs with imperfect channel state information (CSI) bounded by CSI errors.

Chapter 5 extends the concept introduced in Chapter 4 with the aim of reducing the complexity and memory footprint of SLP-DNet. The chapter explicitly investigates the impact of NN weight quantisation, where binary and ternary weight quantisation techniques are introduced to reduce the size of the SLP-DNet model and improve training and inference efficiencies. A stochastic quantisation scheme is presented in which the weight matrix is partitioned, part is quantised, and the other is retained in its full floating-point presentation.

Chapter 6 proposes an efficient and scalable deep neural network-based MIMO detector, where complexity is adjusted at inference with an scalable degradation in the detection accuracy. The chapter describes a weight scaling framework that employs monotonically non-increasing profile functions to prioritise a fraction of the layer weights during training. It also explains how an NN architecture is made to self-adjust to the detection complexity, and the profile functions themselves are made trainable parameters.

Chapter 7 concludes this thesis with a summary of the previous chapters' contributions and discusses potential future extensions of the research within the context of this thesis.

1.5 List of Publications

The above contributions in this thesis have resulted in the following peer-reviewed publications:

Journal Papers:

[J1] **A. Mohammad**, C. Masouros, and Y. Andreopoulos, "Complexity-scalable Neural-Network-based MIMO Detection with Learnable Weight Scaling," *IEEE Transactions on Communications,* vol. 68, no. 10, pp. 6101–6113, 2020.

[J2] **A. Mohammad**, C. Masouros, and Y. Andreopoulos, "An Unsupervised Deep Unfolding Framework for Robust Symbol Level Precoding," *submitted to IEEE Transactions on Signal Processing,* 2021, available online: https://arxiv.org/abs/2111.08129.

[J3] **A. Mohammad**, C. Masouros, and Y. Andreopoulos, "A Memory-Efficient Learning Framework for Symbol Level Precoding with Quantized NN Weights," *submitted to IEEE Transactions on Signal Processing,* 2021, available online: https://arxiv.org/pdf/2110.06542.

Conference Papers:

[C1] **A. Mohammad**, C. Masouros, and Y. Andreopoulos, "Accelerated Learning-based MIMO Detection through Weighted Neural Network Design," *in ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE,* 2020, pp. 1–6.

[C2] **A. Mohammad**, C. Masouros, and Y. Andreopoulos, "An Unsupervised Learning-Based Approach for Symbol-Level-Precoding," *2021 IEEE Global Communications Conference (GLOBECOME) Workshops, accepted,* Spain, 2021.

[C3] **A. Mohammad**, C. Masouros, and Y. Andreopoulos, "Learning-Based Symbol Level Precoding: A Memory-Efficient Unsupervised Learning Approach," *IEEE Wireless Communications and Networking Conference (IEEE WCNC 2022), accepted*, Austin, TX, USA, 2022, available online: https://arxiv.org/abs/2111.08110.

Chapter 2

MIMO Fundamentals and Related Theoretical Concepts

This chapter presents a theoretical overview of the multi-antenna wireless communication system pertinent to this work. It discusses optimal and sub-optimal multi-user precoding and signal detection or equalisation schemes for transmitter and receiver designs in MIMO settings. Most importantly, the chapter explains relevant literature leading to the evolution of precoding/beamforming algorithms from classical approaches based on a block of symbols to the precoding performed on a symbol level basis based on convex optimisation techniques. In addition, MIMO signal detection techniques are briefly reviewed.

2.1 MIMO Communication Systems-Principles

Digital transmission using multiple antenna systems, known as MIMO, is one of the most notable breakthroughs of modern communication systems. MIMO system has been proven to be one of the critical technologies that can resolve the impediment of the traffic capacity in 5G and beyond wireless communications [31–33]. The fundamental concept of the MIMO system is that signals at both transmitting and receiving antennas are fused so that the quality or data rate of the communication links of each user is improved. Through this, QoS and the operator's revenues will significantly increase [31].

In MIMO systems, the main idea is that signal processing is augmented with spatial dimensions integrated using spatially distributed multiple antennas. In particular, with the deployment of multiple antennas at the BS, parallel data streams can be transmitted concurrently to accommodate spatial multiplexing [34]. Th space-time coding techniques are used to send multiple copies of data across the antenna arrays to improve transmission diversity. The successes recorded by the MIMO system as the critical element in implementing new technologies for 5G are due to its ability to improve the performance of communication in order of magnitude without the additional cost of spectrum, but with hardware and algorithm complexities [2].

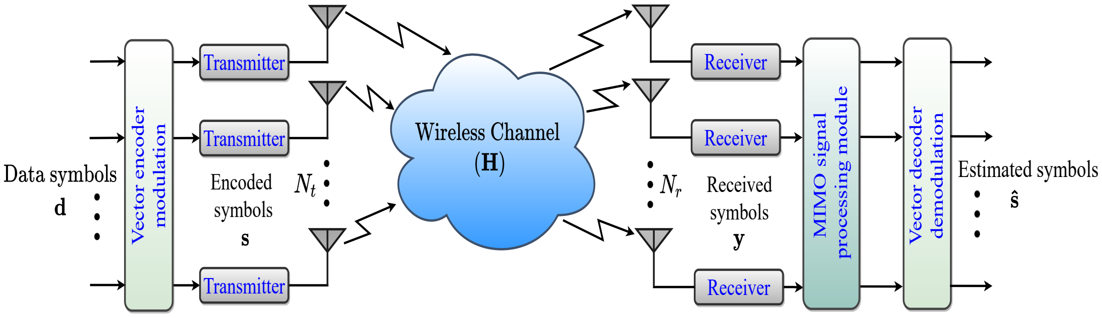


Figure 2.1: A general block diagram of MIMO communication system [34]

A typical MIMO wireless communication system, as depicted in Figure 2.1. The data symbol streams are encoded with a vector encoder denoted by **s** and transmitted simultaneously from N_t BS transmit antennas to a single receiver having N_r receive antennas. The MIMO processing unit at the receiver side estimates the data symbol streams from the received sample based-band signals **y** to produce the estimated data symbols $\hat{\mathbf{s}}$. Generally, a MIMO system having N_t transmit and N_r receive antennas has N_t N_r sub-channels between the transmitter and receiver [2]. Therefore, the each sub-channel is modelled as a linear discrete-time finite impulse response (FIR) with complex coefficients. In a flat fading scenario, the signal in each sub-channel is attenuated and phase-shifted due to propagation delay between the transmit and receive antennas. The sub-channel is thus reduced to a one-tap FIR filter (one complex coefficient). However, the channel becomes quasi-static if it remains constant over the whole transmission time slot [34]. Throughout this thesis, we have assumed a quasi-static flat fading wireless channel.

Mathematically, at each j-th receive antenna, there is n_j additive white Gaussian noise (AWGN) a with zero mean and variance σ^2, the received signal vector in MIMO system can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}, \tag{2.1}$$

where $\mathbf{y} \in \mathbb{C}^{N_r \times 1}$ and $\mathbf{s} \in \mathbb{C}^{N_t \times 1}$ represent the receive and transmit signal vectors, respectively. $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is the Rayleigh fading channel matrix and $\mathbf{n} \in \mathbb{C}^{N_r \times 1}$ is the noise vector obtain from random Gaussian distribution $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{1})$. From (2.1), the received symbol is a linear combination of the transmitted symbol. Each transmitter sends different linear combinations of symbols over the j-th channels. However, the transmitter has no control over the channel but can decide the specific linear combinations of symbols to send by suppressing some signals or aligning them to a particular direction using a pre-processing technique known as precoding [35]. The symbol ready for transmission is $\mathbf{s} = \mathbf{W}\mathbf{d}$, where $\mathbf{W}$ is the transmit precoding matrix. A signal detector is used at the receiver to decode the received signal vector to obtain the estimated output symbols.

2.1.1 Benefits of MIMO Systems

The inherent benefits of multi-antenna systems are array gain, diversity gain and spatial multiplexing gain. These are briefly explained below.

Array gain: Array gain is the increase in receive SNR through a coherent combination of the transmitted signal at the receiver [32]. Array gain is achieved when knowledge of the CSI is known by either the transmitter or receiver. With Array gain, the noise resistance is improved, thus enhancing the coverage and range of a wireless network.

Diversity gain: The received signal level at a receiver fluctuates or fades. Multiple antennas at the transmitter or receiver offer spatial diversity gain that allows signal transmission through several independent fading paths, providing the receiver with multiple copies of the transmitted signal in time, frequency, or space [33]. In this way, the probability that at least one or more copies of the signal does not experience deep fade, thereby enhancing the quality and reliability of signal reception. For the MIMO channel with N_t transmit antennas and N_r receive antenna, there are N_tN_r independent fading links or sub-channels, and is equal to the spatial diversity order [36].

Spatial Multiplexing: Spatial multiplexing refers to using multiple antennas at both the transmitter and receiver to transmit multiple data streams simultaneously within the same frequency band to increase information capacity [33].

One of the principal metrics used to evaluate wireless communication systems' performance is bit-error rate (BER). BER is the ratio of the number of received bits that have been changed while passing through the communication channel to the number of bits sent, defined as

$$BER = \frac{n_e}{n_b}, \tag{2.2}$$

where n_e and n_b are the number of erroneous bits and total number of transmitted bits. In this thesis, we use BER as the performance metric as we shall see in Chapter 4. The approximate BER for a MIMO system with N_r receive antennas is expressed as [34]

$$BER = \binom{2N_r - 1}{N_r} \left(\frac{1}{2\rho}\right)^{N_r}, \tag{2.3}$$

where $\binom{2N_r-1}{N_r}$ denotes the number of combinations of selecting N_r antennas from the set of $2N_r - 1$ antennas. An equaliser is needed to decode the transmitted symbol vector $\mathbf{s}$ and manage inter symbol interference (ISI) at the receiver. We exclusively assume perfect channel CSI at the receiver in the subsequent subsections and discuss different MIMO equalisation schemes relevant to this work.

2.2 Channel Modelling

As we shall see later, the channel's knowledge is required for signal preprocessing at the transmitter. Similarly, the receiver needs to also know the CSI for signal post-processing to design signal detectors that can efficiently decode the received symbols. Therefore, channel modelling is essential in analysing and designing precoders and signal detectors or equalisers. Various statistical distributions have been used to model the random change in multipath. The most typical of these are the Rayleigh and Rician models [37]. The Rician distribution is used to model a wireless channel when a direct line-of-sight (LoS) between the transmitter and receiver exists [38]. In this work, we adopt the Rayleigh channel because it describes the form of fading due to multipath propagation when there is no dominant signal (i.e. LoS). It is also a typical channel model widely adopted to the model radio environment [39].

2.2.1 Rayleigh Distribution

Rayleigh fading channel is a statistical distribution for non LoS communication channels [40]. In terms of base-band representation, the channel h is defined as a random variable in complex domain as follows

$$h = h_R + jh_I, \tag{2.4}$$

where h_R and h_I are the real and imaginary components of the channel response, respectively. When the signal arrives at the receiver from various paths of nearly equal power, the resulting field is the sum of real and imaginary parts of sums of identically distributed random variables. By using Central Limit Theorem [41], the summation of the identically distributed random variables obeys a Gaussian distribution with zero mean and variance σ_s^2 (i.e. channel's average power).

Uncorrelated Rayleigh channel: Rayleigh distribution considers the presence of large-scale statistically independent reflectors and scatters in the wireless radio space; then, each path of the channel tap can be modelled as a complex random quantity [40]. Consequently, an uncorrelated MIMO channel coefficient can be modelled as

$$\mathbf{H} \sim \mathcal{N}(0,\ 1) \in \mathbb{C}^{N_r \times N_t}. \tag{2.5}$$

Correlated Rayleigh channel: The channels between neighbouring antennas are correlated in a practical antenna array, mainly when the antenna separation is less than the carrier wavelength [39]. This is contrary to the uncorrelated channel model that ignores the spatial correlation effect between the antenna elements.

2.3 MIMO Downlink Transmission

Throughout this thesis, the channel model we use is based on the general MIMO channel model and the descriptions for MU-MIMO systems.

An extra signal processing at both transmitter and receiver is required in a multi-antenna system, contrary to the traditional single-antenna systems. In this context, depending on which side the processing is applied, the signal processing can be precoding schemes at the transmitter side and detection methods at the receiver side [42]. The signal processing can be classified into precoding schemes at the transmitter and detection techniques at the receiver, depending on which side the processing is applied. In a MU-MIMO system, combined signal processing of data streams is usually challenging for several users in the downlink due to the gap in physical areas. While precoding is favoured in the downlink transmission, receive combining techniques are usually employed at the receiver side or user end [43]. With the knowledge of the channel at the BS, precoding can relieve the computational load of the users by transferring the signal processing process from the user side to the BS [43]. And this is what makes the precoding techniques uniquely popular and most widely investigated.

2.3.1 MU-MIMO Channel Model

A multi-user MIMO system is illustrated in Figure 2.2 with N_t transmit antennas at the BS serving N_r receive antennas at k-th user, where $N_r = \sum_{k=1}^{K} N_k$ and $\mathbf{W}_k$ denotes the precoding matrix. Here, we imagine the BS is separated from its rich scattering environment. From the transmitter's perspective, the channel's spatial

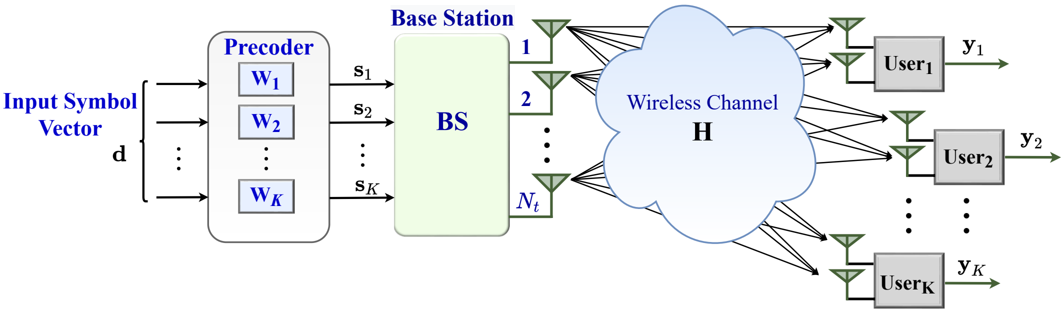


Figure 2.2: MU-MIMO Channel Model

configuration is now influenced by remote scattering objects resulting in a highly spatially correlated situation with a small number M of dominant far off scattering objects. Hence, the channel is expressed as [44, 45]

$$\mathbf{H} = \sqrt{\frac{N_r}{\mathrm{tr}(\mathbf{A}\mathbf{A}^H)}}\mathbf{G}\mathbf{A}^H, \tag{2.6}$$

where $\mathbf{A} \in \mathbb{C}^{N_r \times M}$ is the antenna steering matrix containing M vectors of the transmit antenna response having M directions of departure (DoD) and $\mathbf{G} \sim \mathcal{N}(0,\ 1) \in \mathbb{C}^{N_t \times M}$. In this situation, the channel is viewed as semi-correlated, where the spatial correlation solely exists at the transmit side [45]. For a uniform linear arrays (ULAs), each vector in $\mathbf{A}$ is modelled as

$$\mathbf{A}_k = \frac{1}{\sqrt{M}} \cdot \left[\mathbf{a}^H(\phi_{k,1}), \cdots, \mathbf{a}^H(\phi_{k,M})\right]^H, \ \forall k;\ k = 1, \ldots, N_r, \tag{2.7}$$

where $\mathbf{a}(\phi_{k,i}) = \left[1, e^{j2\pi r \sin\phi_{k,i}}, \cdots, e^{j2\pi(N_r-1)r\sin\phi_{k,i}}\right] \in \mathbb{C}^{M\times 1}$, $\forall k;\ k = 1, \ldots, N_r$. The r here represents the normalised antenna spacing (normalised by the carrier wavelength), and $\phi_{k,i}$ is the steering angle and is assumed to obey a Laplacian distribution [45]. Following the above, the resultant channel matrix of the system model is

$$\mathbf{H} = [\mathbf{H}_1, \cdots, \mathbf{H}_K] \ \in \mathbb{C}^{N_r \times N_t},\ \forall k;\ k = 1, \ldots, N_r, \tag{2.8}$$

where $\mathbf{H}_k \in \mathbb{C}^{N_t \times N_k}$ is the k-th user' channel matrix. However, in multi-user multiple-inputs-single-output (MU-MISO) systems, the combined channel matrix is composed of the channel vectors of each user, and is given by

$$\mathbf{H} = [\mathbf{h}_1, \cdots, \mathbf{h}_K] \ \in \mathbb{C}^{N_r \times N_t},\ \forall i;\ i = 1, \ldots, K, \tag{2.9}$$

where $\mathbf{h}_i \in \mathbb{C}^{N_t \times 1}$ is the i-th user' channel matrix.

2.3.2 Imperfect CSI Modelling

The knowledge of the channel is required at the BS for downlink precoding designs, as earlier explained. Generally, acquiring a perfect knowledge of the channel is difficult in practical wireless communication systems. Because of this, it is imperative to analyse the performance of downlink transmission techniques under imperfect CSI. This subsection presents the imperfect channel model used in the subsequent chapters.

2.3.2.1 Model of Statistical CSI Error

The uplink and downlink channels operate in the same frequency bands in the time division duplex (TDD) mode. This allows the downlink channel to be directly measured at the BS by the uplink-downlink channel reciprocity as a function of an estimation error [46]. As described in [47], the imperfect channel model is given by

$$\mathbf{H} = \kappa \cdot \left(\tilde{\mathbf{H}} + \tilde{\mathbf{E}}\right) + \mathbf{R}, \tag{2.10}$$

where $\mathbf{H}$ is the actual channel matrix, $\tilde{\mathbf{H}}$ denotes the estimated channel at the BS and $\tilde{\mathbf{E}}$ is the estimation error and κ is the correlation factor associated with the channel estimation time delay. $\mathbf{R}$ designates the delay error matrix, whose entries are i.i.d. $\mathcal{N}(0,\ 1-\kappa^2)$. A simplified imperfect CSI model for TDD transmission is obtained when $\kappa = 1$, and is expressed as

$$\mathbf{H} = \tilde{\mathbf{H}} + \tilde{\mathbf{E}}. \tag{2.11}$$

2.3.2.2 Norm-Bounded CSI Error Model

In frequency division duplex (FDD) mode, the downlink uplink channel transmissions are performed simultaneously using different frequency bands. Because of this, the uplink-downlink reciprocity does not exist. Practically, the estimation of the channel knowledge is first performed at the receivers and then feedback to the BS [46]. Therefore, for FDD systems, the CSI errors are dominated by the quantisation errors in the limited feedback. In this context, the imperfect channel is modelled as [46, 48]

$$\mathbf{h}_i = \tilde{\mathbf{h}}_i + \tilde{\mathbf{e}}_i,\ \forall i \in \{1, \cdots, K\}. \tag{2.12}$$

It is important to note that the channel uncertainty is bounded bounded by a spherical region for each user, and is expressed as [49]

$$\xi = \left\{ \tilde{\mathbf{h}}_i + \tilde{\mathbf{e}}_i |_{\|\tilde{\mathbf{e}}_i \leq 1\|} \right\}, \; \forall i \in \{1, \cdots, K\}. \tag{2.13}$$

2.4 Precoding

Despite the MIMO systems' performance benefits and increased spectral efficiency, high power consumption and computational complexity of the MIMO decoding techniques have rendered receive processing practically implausible at the user equipment (UE) [42, 50]. The UE, such as mobile handsets, are typically limited to simple low computational complexity algorithms. Therefore, to sustain a simple and cost-effective UE, the complex and power-consuming signal processing is shifted to the BS for downlink transmission, via a technique called precoding [50]. Precoding is a signal processing technique that exploits CSI at the BS or transmitter applied on a data symbol before transmission [42].Various precoding methods for downlink transmission have been reported in the literature, from complex but high-performance non-linear precoding techniques [43, 51–60] to low complexity linear precoding methods [5, 50, 61, 62]. Similarly, many optimisation-based precoding techniques that exploit CI [49, 58, 60, 63–65] based on convex optimisation theory have also been put forward and will be discussed in the subsequent subsections.

2.4.1 Linear Precoding

Linear precoding is a set of simple transmission strategies, where the data symbols to be transmitted are combined linearly with the precoding matrix **P** to produce the precoded signal vector **s** before the transmission. Linear precoding approaches are most appealing due to their simplicity and low computational complexity with poor performance compared to their nonlinear counterparts. The generic expression for a precoded signal is given by

$$\mathbf{s} = \mathbf{P}\mathbf{d} = \frac{1}{f} \cdot \mathbf{W}\mathbf{d} \tag{2.14}$$

from (2.14), $f = \sqrt{\mathrm{tr}\left(\mathbf{W}\mathbf{W}^H\right)}$ represents a scaling factor, $\mathbf{P} \in \mathbb{C}^{N_t \times K}$ and $\mathbf{W} \in \mathbb{C}^{N_t \times K}$ are normalised and non-normalised precoding matrices, respectively. Due to uplink-downlink reciprocity [5], the downlink channel in the TDD mode is typically assumed to be the conjugate transpose or *'Hermitian'* of the channel matrix $\mathbf{H}$ of the uplink. Therefore, the received signal vector at K UEs is

$$\mathbf{y} = \mathbf{H}^H\mathbf{s} + \mathbf{n} = \frac{1}{f} \cdot \mathbf{H}^H\mathbf{W}\mathbf{d} + \mathbf{n}, \tag{2.15}$$

where $\mathbf{n} \in \mathbb{C}^{K \times 1}$ represent noise interference. In the subsequent sub-subsections, we present typical linear precoding schemes with their closed-form expressions.

2.4.1.1 Match Filter Precoding (MF-P)

MF precoder also known as maximum ratio transmission (MRT) is the most rudimentary precoding technique, which disregards the MUI while maximising the received SNR. Mathematically, the match filter precoder is modelled as the *'Hermitian'* of the channel matrix $\mathbf{H}$ [62]

$$\mathbf{P}^{MFP} = \frac{1}{f^{MFP}} \cdot \mathbf{H} = \frac{\mathbf{H}}{\sqrt{\mathrm{tr}\left(\mathbf{H}^H\mathbf{H}\right)}}. \tag{2.16}$$

The received signal vector is given by

$$\mathbf{y}^{MFP} = \frac{\mathbf{H}^H\mathbf{H}}{\sqrt{\mathrm{tr}\left(\mathbf{H}^H\mathbf{H}\right)}} \cdot \mathbf{d} + \mathbf{n}. \tag{2.17}$$

For massive MU-MIMO scenarios, MF-P can provide potential complexity gains, while its performance suffers in interference-limited scenarios.

2.4.1.2 Zero-Forcing Precoder (ZF-P)

The ZF-P technique has been widely investigated because of its simple structure [62, 66]. The precoding matrix is obtained as

$$\mathbf{P}^{ZFP} = \frac{1}{f^{ZFP}} \cdot \mathbf{H}\left(\mathbf{H}^H\mathbf{H}\right)^{-1} = \frac{\mathbf{H}\left(\mathbf{H}^H\mathbf{H}\right)^{-1}}{\sqrt{\mathrm{tr}\left[\left(\mathbf{H}^H\mathbf{H}\right)^{-1}\right]}}, \quad \text{for } N_t \geq K, \tag{2.18}$$

Thus, the equivalent received signal vector is

$$\mathbf{y}^{ZFP} = \frac{\mathbf{H}^H\mathbf{H}\left(\mathbf{H}^H\mathbf{H}\right)^{-1}}{\sqrt{\operatorname{tr}\left[\left(\mathbf{H}^H\mathbf{H}\right)^{-1}\right]}} \cdot \mathbf{d} + \mathbf{n}. \tag{2.19}$$

Contrary to MF-P, which has low performance at high SNR, ZF-P offers an improved performance gain over the MF-P at a high SNR regime.

2.4.1.3 Regularised Zero-Forcing Precoder (RZF-P)

The direct channel inversion, as in ZF-P, generally leads to poor performance due to the singular value spread of the channel matrix [53]. Channel regularisation factor is employed to deal with the problem of the ill-conditioned channel. The RZF precoding matrix is given by

$$\mathbf{P}^{RZFP} = \frac{1}{f^{RZFP}} \cdot \mathbf{H}\left(\mathbf{H}^H\mathbf{H} + \beta\mathbf{I}\right)^{-1} = \frac{\mathbf{H}\left(\mathbf{H}^H\mathbf{H} + \beta\mathbf{I}\right)^{-1}}{\sqrt{\operatorname{tr}\left[\left(\mathbf{H}^H\mathbf{H} + \beta\mathbf{I}\right)^{-1}\mathbf{H}^H\mathbf{H}\left(\mathbf{H}^H\mathbf{H} + \beta\mathbf{I}\right)^{-1}\right]}}, \tag{2.20}$$

where $\beta = K\sigma_n^2$ is the regularisation factor and σ_n^2 is the noise power at the receiver. When $\sigma_n^2 = 0$, (2.20) reduces to (2.18). The amount of interference can be controlled by by setting $\beta > 0$. The received signal vector is obtained as

$$\mathbf{y}^{RZFP} = \mathbf{H}^H\mathbf{P}^{RZFP} \cdot \mathbf{d} + \mathbf{n}. \tag{2.21}$$

It is important to note that RZF-P has almost equal computational cost as the ZF-P scheme [66]. The RZF-P scheme is usually referred to as Minimum Mean Square-Error (MMSE) precoding.

2.4.2 Non-Linear Precoding

As earlier mentioned, linear precoding techniques generally have simple closed-form structures making them relatively computationally efficient. However, their performance is far from the optimal theoretical capacity. Several non-linear precoding techniques have been proposed in the literature to decrease this performance gap. The first theoretical precoding scheme known to achieve the sum-rate capacity is dirty paper coding (DPC) [51]. Furthermore, many other remarkable contributions have also been reported within the scope of multi-antenna DPC, including Vertical-Bell Laboratories Layered Space-Time (V-BLAST) precoding [67], Tomlinson-Harashima Precoding (THP) [43] , nested lattice [68], trellis precoding [68] and vector perturbation (VP) [54]. As our focus in this Thesis is on optimisation based precoding, in the following we focus on this category of precoding techniques.

2.5 Optimisation-Based Methods

In recent years, convex optimisation has been successfully and widely applied to various problems in signal processing and other science and non-science related problems. Specifically, many optimisation-based algorithms have been designed to deliver optimum solutions in physical layer wireless communications, such as full-duplex MIMO, energy harvesting, precoding, signal detection, and multicell coordinated beamforming, etc. Consequently, many optimisation-based beamforming techniques have been introduced in the literature and can be classified into two: block level precoding (BLP) and symbol level precoding (SLP) schemes.

2.5.1 Optimisation-based Precoding

This type of optimisation-based precoding is based on the traditional approach that treats interference as harmful. Several precoding schemes that fall into this category for perfect CSI [69–73] and imperfect CSI cases [74–77] have been reported in the literature based on the desired system performance metrics:

1. **Power Minimisation Problem:** Conventionally, the power minimisation problem seeks to minimise the average transmit power by treating all interference as detrimental subject to the users' QoS signal-to-interference-noise ratio (SINR) constraints is expressed as [73]

$$\begin{aligned} \min_{\{\mathbf{w_i}\}} \quad & \sum_{i=1}^{K} \|\mathbf{w}_i\|^2 \\ \text{s.t.} \quad & \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{i=1,k\neq i}^{K} |\mathbf{h}_k^H \mathbf{w}_i|^2 + \sigma_n^2} \geq \Gamma_k \text{ (minimum required QoS)}, \ \forall k \in \{1,\cdots,K\}, \end{aligned} \tag{2.22}$$

where Γ_k is the minimum required QoS of the k-th user that produces the precoding vectors that yield the minimum transmit power. It has been proven that problem (2.22) is sub-optimal from an instantaneous point of view, as it does not take into account the fact that interference can constructively enhance the received signal power [61]. This problem is a typical block level precoding based on power minimisation formulation.

2. **SINR-balancing problem:** This involves maximising the minimum SINR subject to the total power constraints.This problem is mathematically formulated as

$$\begin{aligned} &\max_{\mathbf{w_i},}\ \min_{\boldsymbol{\gamma}_\mathbf{k}} && \gamma_k \\ &\text{s.t.} && \gamma_k = \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{i=1,k\neq i}^{K} |\mathbf{h}_k^H \mathbf{w}_i|^2 + \sigma_n^2},\ \forall k\ \in \{1,\cdots,K\}, \\ &&& \sum_{i=1}^{K} \|\mathbf{w}_i\|^2 \leq P_{\text{total}}, \end{aligned} \tag{2.23}$$

where P_{total} denotes the total transmit power available at the BS.

3. **Sum-Rate Maximisation:** This optimisation problem indirectly seeks to maximise the communication system's spectral efficiency subject to total power constraints, and mathematically formulated as [78]

$$\begin{aligned} &\underset{\mathbf{w_i}}{\text{maximise}} && \sum_{k=1}^{K} \log\left[1 + \frac{\mathbf{h}_k^H \mathbf{w}_k}{\sum_{i\neq k}^{K} \mathbf{h}_k^H \mathbf{w}_i + \sigma_{nk}^2}\right] \\ &\text{s.t.} && \sum_{i=1}^{K} \|\mathbf{w}_i\|^2 \leq P_{\text{total}},\ \forall k\ \in \{1,\cdots,K\}. \end{aligned} \tag{2.24}$$

Another variant of this formulation is the weighted sum rate (WSR), which can be solved as a weighted minimum-mean squared error (WMMSE) problem with an optimised mean-squared error (MSE)-weights [79].

Other optimisation problems in this category are leakage based precoding, error rate minimisation, etc. Among the three optimisation problems discussed in Subsection 2.5.1, transmit power minimisation is particularly important because it

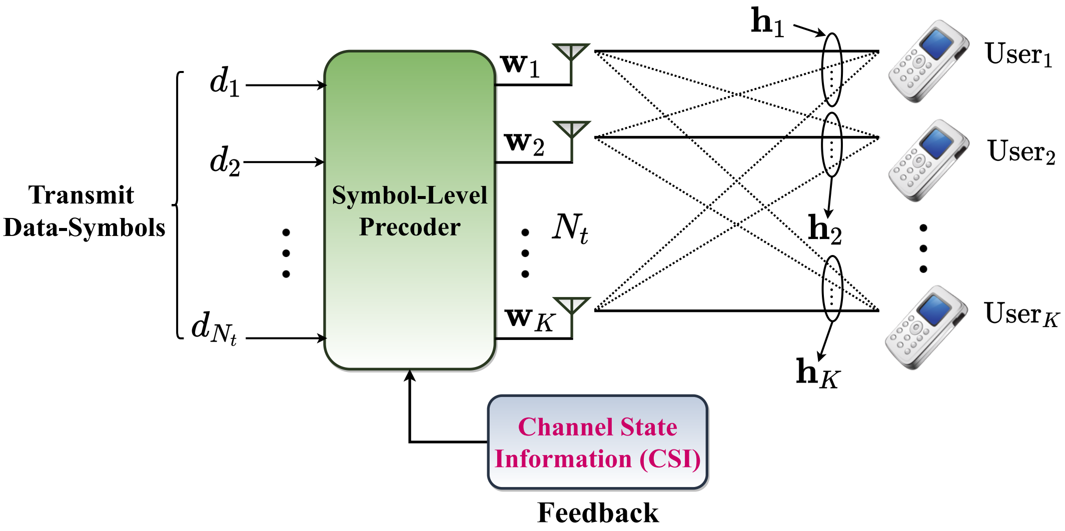


Figure 2.3: Generic structure of symbol level precoder

directly addresses the bottleneck associated with the power efficiency of wireless transmission links. Optimal power transmission strategies provide communications by reducing CO_2 emission and the operator's operational expenditures. Therefore, in this thesis, we focus on designing learning-based solutions for optimal power minimisation problems.

2.5.2 Constructive Interference Optimisation-based Precoding

The precoding methods discussed in the previous sections apply the precoding coefficients across the block of symbols or codewords, and hence they are classified as block-level precoding (BLP) schemes. This means that the precoding matrices do not depend on the data symbols. We will, in this subsection, discuss the precoding techniques that exploit multiuser interference, where the precoding coefficients are applied on a symbol basis. These types of precoding schemes are termed symbol level precoding (SLP). Figure 2.3 shows a typical SLP setting for MU-MISO downlink transmission with N_t BS antennas serving K users. The BS is assumed to know the channel through CSI feedback acquired from the user end. The solid lines from each transmitter represent the non-interfering signal towards the desired users, while the dotted lines are the co-channel interference from unintended users. Instead of suppressing the multiuser interference as in BLP, we introduce the concept of constructive interference (CI), which enables the precoding schemes to exploit the instantaneous interference and transform it useful signal to enhance signal detection at the receiver.

Suppose the desired user is user1, then through CI, the interfering signals from other users can be added constructively to enhance the received signal power of the user1. Therefore, precoding matrix is expressed as

$$\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_K] \ \in \mathbb{C}^{N_t \times K}, \ \forall k; \ k = 1, \ldots, K \tag{2.25}$$

where $\mathbf{w}_k \in \mathbb{C}^{N_t \times 1}$ is the k-th user vector. In this type of precoding, the precoding coefficients are applied on a symbol basis. CI is defined as the interference that forces the received signals beyond the modulation constellations' detection boundaries or thresholds [63].

With the aim of utilising the instantaneous interference in a multi-user downlink channel scenario, the interference can be categorised into constructive and destructive based on the known standards described in [80, 81]. An initial closed-form symbol-level-assisted linear precoding optimisation that harnesses the CI while annulling the destructive part was proposed in [63]. In this case, the instantaneous interference can contribute constructively to the detection of the desired signal. Knowing both the data symbols and the CSI at the BS, we can transform the SINR constraints in (2.22) and (2.23) to include CI for generic $\mathcal{M}$-array phase-shift keying ($\mathcal{M}$-PSK) modulated signals. This proposition is depicted in Figure 2.4, showing the constructive interference and destructive interference regions for quadrature phase-shift keying (QPSK) and 8PSK constellation points, where the green areas represent the constructive region. As observed from Figure 2.4, τ is the distance between the nominal constellation point and the decision variable of the constellation, $\mathrm{x}_{re} = \Re\{\tilde{\mathbf{y}}\}$ and $\mathrm{x}_{im} = \Im\{\tilde{\mathbf{y}}\}$ are the real and imaginary parts of the phase rotated received signal $\tilde{\mathbf{y}} \triangleq \mathbf{h}_k^H \sum_{k=1}^{K} \mathbf{w}_k e^{j(\varphi_i - \varphi_k)}$ and φ_i is the phase of the desired symbol, respectively. The angle that determines the signal's maximum phase rotation in the constructive region for $\mathcal{M}$ modulation index is given by

$$\phi = \pm \frac{\pi}{\mathcal{M}}. \tag{2.26}$$

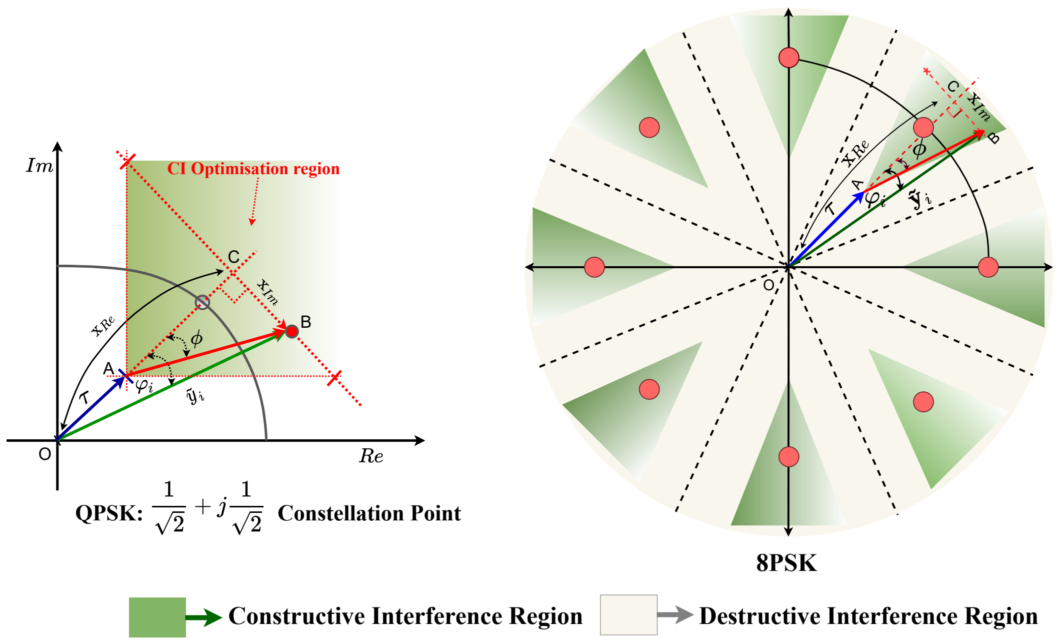

Im
CI Optimisation region
C
x_{Im}
B
x_{Re}
A
ϕ
φ_i
τ
$\tilde{y}_i$
O
Re
QPSK: $\frac{1}{\sqrt{2}} + j\frac{1}{\sqrt{2}}$ Constellation Point
8PSK
Constructive Interference Region
Destructive Interference Region

Figure 2.4: Constructive interference regions for QPSK and 8PSK constellation points [49]

It is also important to note that the instantaneous interference is said to be harnessed constructively if the received signal falls within the green area based on the minimum distance (τ) from the decision boundaries. This allows the interfering signals to align with the symbol of interest constructively, contributing to the desired signal's strength. The x_{Im} and x_{Re} are the imaginary parts of the noiseless received signal $\tilde{y}$. Therefore, by using the geometry, the following vectors can be expressed as

$$\vec{AC} = [\mathrm{x}_{Re} - \vec{OA}], \; \vec{BC} = j \cdot \mathrm{x}_{Im}, \tag{2.27}$$

$\vec{OA}$ is the detection threshold (τ) and is determined from the relation

$$\vec{OA} = \sqrt{\Gamma_i \sigma_n^2}. \tag{2.28}$$

For a point B to be located in the constructive region, the following condition must hold

$$\tan\varphi_i \leq \tan\phi \Rightarrow \frac{|j \cdot \mathrm{x}_{Im}|}{\left|[\mathrm{x}_{Re} - \vec{OA}]\right|} \leq \tan\phi, \tag{2.29}$$

$$\mathrm{x}_{Im} \leq (\mathrm{x}_{Re} - \tau)\tan\phi. \tag{2.30}$$

The CI for each user is guaranteed by adopting (2.30) as the SINR constraint. Consequently, the optimisation problems in (2.22) and (2.23) can be transformed into their equivalent CI-based optimisation formulations as follows [49]

- **Power minimisation CI-based optimisation:**

$$\begin{aligned} \min_{\{\mathbf{w_k}\}} \quad & \sum_{k=1}^{K} \|\mathbf{w}_k d_k\|^2 \\ \text{s.t.} \quad & \left| \Im\left(\mathbf{h}_i^H \sum_{k=1}^{K} \mathbf{w}_k e^{j(\varphi_k - \varphi_i)} \right) \right| \leq \\ & \left(\Re\left(\mathbf{h}_i^H \sum_{k=1}^{K} \mathbf{w}_k e^{j(\varphi_k - \varphi_i)} \right) - \sqrt{\Gamma_i n_0} \right) \tan\phi, \; \forall k \; \in \{1, \cdots, K\}, \end{aligned} \tag{2.31}$$

where $\mathrm{x}_{Im} = \Im\left(\mathbf{h}_i^H \sum_{k=1}^{K} \mathbf{w}_k e^{j(\varphi_k - \varphi_i)} \right)$ and $\mathrm{x}_{Re} = \Re\left(\mathbf{h}_i^H \sum_{k=1}^{K} \mathbf{w}_k e^{j(\varphi_k - \varphi_i)} \right)$. It can be observed that (2.31) is data-dependent; therefore, the optimisation is done on a symbol-by-symbol basis, and such precoding is termed symbol level precoding (SLP).

2.5.3 Robust Power Minimisation Bounded with CSI Errors

The exact CSI is often unobtainable in practice. To model the user's actual channel in the uncertainty region, we consider an ellipsoid ξ such that the channel error is within the uncertainty region of the ellipsoid (i.e $\hat{\mathbf{h}}_i \in \xi$). The model of the uncertainty ellipsoid with the centre $\hat{\mathbf{h}}_i$ is expressed as [49]

$$\xi = \left\{ \hat{\mathbf{h}}_i + \hat{\mathbf{e}}_i |_{\|\hat{\mathbf{e}}_i \leq 1\|} \right\}, \tag{2.32}$$

As shown in [49], the channel error is given by $\left\{ \hat{\mathbf{e}}_i : \|\hat{\mathbf{e}}_i\|_2^2 \leq \varsigma_i^2 \right\}$. It is important to note that the BS is assumed to have the knowledge about the channel error, excluding its corresponding error bound ς_i^2. Given this, the conventional robust precoding for the downlink MU-MISO power minimisation optimisation is [77]

$$\begin{aligned} \min_{\{\mathbf{w_i}\}} \quad & \sum_{i=1}^{K} \|\mathbf{w}_i\|^2 \\ \text{s.t.} \quad & \frac{|\mathbf{h}_i^H \mathbf{w}_i|^2}{\sum_{k=1,k\neq i}^{K} |\mathbf{h}_i^H \mathbf{w}_k|^2 + n_0} \geq \Gamma_i \ \forall e_i, \in \mathcal{U}_I, \ \forall k, \end{aligned} \tag{2.33}$$

The robust beamforming problem (2.33) is nonconvex and can therefore be relaxed to its equivalent semi-definite programming (SDP) problem below

$$\begin{aligned} \min_{\{\bar{\mathbf{W}}_i \succeq 0,\ d_i \geq 0\}} \quad & \sum_{i=1}^{K} \text{trace}(\bar{\mathbf{W}}_i) \\ \text{s.t.} \quad & \begin{bmatrix} \hat{\mathbf{h}}_i^* T_i \hat{\mathbf{h}}_i^T - \gamma_i n_0 - d_i \varsigma_i^2 & \hat{\mathbf{h}}_i^* T_i \\ T_i \hat{\mathbf{h}}_i^T & T_i + \varsigma_i^2 \mathbf{I} \end{bmatrix} \succeq 0 \ \forall k \end{aligned} \tag{2.34}$$

where $T_i \triangleq \bar{\mathbf{W}}_i - \Gamma_i \sum_{k=1,k\neq i}^{K} \bar{\mathbf{W}}_k \ \forall k$ and $\bar{\mathbf{W}}_i = \mathbf{w}_i \mathbf{w}_i^\dagger$.

Over the last decade, there have been tremendous performance gains through interference exploitation based on symbol level optimisation for PSK and QAM (quadrature amplitude modulation) modulated signals [14–16, 21, 82–85]. However, the optimisation solutions are often based on traditional mathematical formulation, which may sometimes not be tractable or too complex to solve due to the problem's dimension. Albeit providing significant performance benefits compared to linear precoding techniques, non-linear precoding methods involve sophisticated signal processing at the transmitter, making their implementation practically impossible for massive MU-MIMO systems. Therefore, in this thesis, we propose low complexity learning-based precoding solutions in chapters 4-5.

2.5.4 DL-based MIMO Precoding Schemes

Several learning-based precoding/beamforming schemes have been proposed to address the problem of computational complexity. This subsection presents the overview of some learning-based precoding methods, which is the main focus of this thesis, as we shall see in Chapter 4 and Chapter 5.

More relevant to this work are the learning-based precoding schemes for MU-

MISO downlink transmission [24, 26, 27, 30, 86]. The benefit of using DNN is that the computational burden of the learning algorithm can be controlled via online training, and a variety of loss functions can be used for each optimisation objective. One of the earliest attempts of using DNN models for beamforming design was the work of Alkhateeb *et al.* [28], where a learning-based coordinated beamforming technique was proposed for link reliability and frequent poor hand-off between BSs in millimetre-wave (mm-Wave) communications. Kerret and Gesbert [29] introduced DNN precoding scheme to address the *"Team Decision problems"* for a decentralised decision making in MIMO settings. Huang *et al.* [86] proposed a fast beamforming design based on unsupervised learning that yielded performance close to that of the weighted minimum mean-square error (WMMSE) algorithm. A DNN-based precoding strategy that utilised a heuristic solution structure of the downlink beamforming was proposed by Huang *et al.* [30]. Furthermore, Xia *et al.* [24] developed deep convolutional neural networks (CNNs) framework for downlink beamforming optimisation. The framework exploits expert knowledge based on the known structure of optimal iterative solutions for sum-rate maximisation, power minimisation, and SINR balancing problems.

DNN methods are typically used for unconstrained optimisation problems. Therefore, most of the DNN-based strategies for wireless physical layer designs are based on supervised learning to approximate the optimal solutions. Using such approaches, the constraints are implicitly contained in the training dataset obtained from conventional optimisation solutions. However, if obtaining optimal solutions via traditional optimisation methods is very computationally expensive (or infeasible), using DNN for model approximation may not be practical. Furthermore, the common approach for solving constrained optimisation with DNN for wireless physical layer design is via function approximation. It involves solving the problem, first using iterative algorithms or convex optimisation techniques, and finally approximating the optimal solution with a DNN architecture [24, 27, 30]. Table 2.1 summarises the recent milestones in the development of ML-based precoding/beamforming methods. As we can see, significant works have been done on

Table 2.1: Evolution of learning-based precoding/beamforming MIMO schemes

Learning method	Hybrid Precoding	BLP	SLP	Reference
Proposed supervised learning for sum-rate maximisation using DNN as a function approximator		✓		[87]
Proposed supervised learning for sum-rate maximisation by approximating WMMSE solution using DNN		✓		[27]
Proposed an unsupervised learning scheme for sum-rate maximisation		✓		[86]
Proposed an unsupervised learning beamforming technique for sum-rate maximisation		✓		[30]
Proposed supervised learning for physical layer security using a support vector machine (SVM)		✓		[88]
Proposed supervised learning method for sum-rate maximisation and user scheduling using DNN architecture		✓		[89]
Proposed a supervised learning approach to solve power minimisation, SINR balancing and sum-rate maximisation problems		✓		[24]
Proposed a supervised learning technique for Finite-alphabet precoding	✓	✓		[12]
Proposed supervised learning method using deep CNN for sum-rate maximisation		✓		[90]
Proposed deep reinforcement learning approach for sum-rate maximisation in an mm-wave MIMO system		✓		[91]
Proposed a deep auto-encoder (DNN) for End-to-End communications			✓	[92]
Proposed a unique received signal strength indicators (RSSI)-based unsupervised learning technique for hybrid beamforming	✓			[93]
Proposed a supervised learning scheme using a CNN for sum-rate maximisation		✓		[94]
Proposed an unsupervised learning and supervised learning DNN frameworks for sum-rate maximisation in a multi-cell scenario		✓		[95]
Proposed deep unfolding-based WMMSE algorithm integrating expert knowledge for sum-rate maximisation		✓		[96]

learning-based BLP methods, with few literature on SLP learning approach. While many optimisation problems can be addressed in learning-based precoding designs, most research works in this context focus on sum-rate maximisation via supervised learning [24, 27]. The reason is due to the relative simplicity of the optimisation objective function that can be easily unfolded into learning layers or by simply model approximation using DNN architectures. Therefore, there is a gap in designing DL strategies for SLP to solve more challenging optimisation problems, such as power minimisation problem, SINR balancing problem, secrecy problem, etc. Accordingly, the major drawback of these proposals is that the efficacy of the supervised learning is bounded by the assumptions and accuracy of the optimal solutions obtained from the structural optimisation algorithms.

2.6 MIMO Detection Techniques

In section **2.4**, we have discussed the signal processing techniques that are performed at the transmitter prior to the transmission.This section will focus on the post signal processing methods for receiver design (detection or signal equalisation techniques).

According to Claude Shannon, the main objective of communication systems is the efficient recovery of the transmitted symbols at the receiver exactly or as closely as possible as it was originally sent by the transmitter [97]. However, due to channel and physical impairments, it is difficult to decode the transmitted symbols precisely. In MIMO systems, multiple interfering symbols are transmitted concur-

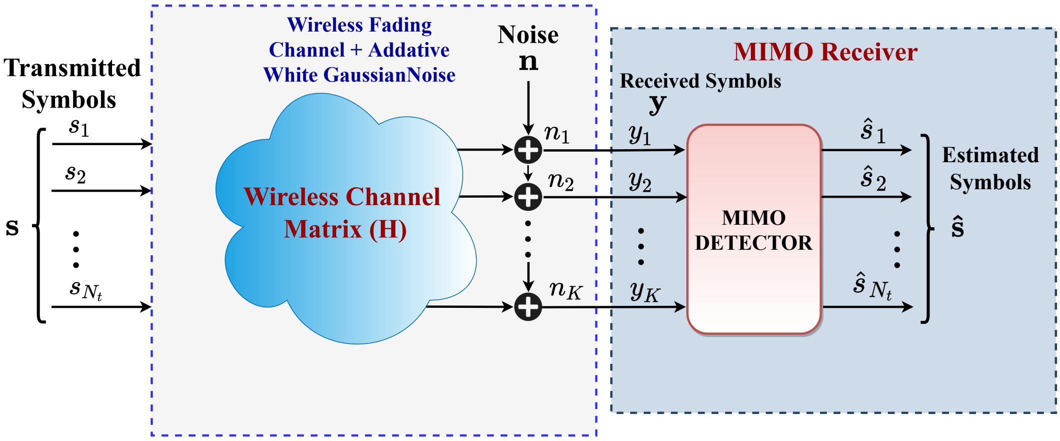


Figure 2.5: Conceptual framework for MIMO Detection problem

rently. These symbols are supposed to be detected/decoded at the receiver contingent on the level of random noise or interfering signals, as shown in Figure 2.5. The task of a MIMO detector is to determine the estimates of the transmitted vector (**s**) from the received vector (**y**) with the lowest minimum error probability. It is possible to detect the transmitted multiple symbols separately or jointly. Contrary to separate detection, each symbol is decoded considering the other symbols' characteristics in joint detection [2]. Moreover, joint detection of multiple symbols in MIMO systems is central to actualising the substantial benefits of different MIMO techniques; and it performs better than separate detection but with additional computational complexity [2, 98]. In the following subsections, we will review some MIMO detection techniques that are relevant to this thesis.

2.6.1 Optimum Detectors

The maximum likelihood detector (ML-D) and maximum *'a posteriori'* detector (MAP-D) are optimal algorithms for solving MIMO detection problems [2]. MAP uses Bayesian inference optimum decision criterion to minimise error probability based only on the observed signals (received symbols). MAP based MIMO detector is formulated as

$$\mathcal{D}^{MAP} : \hat{\mathbf{s}} = \underset{\mathbf{s}\,\in\,\mathbf{S}}{\text{arg max}} \quad Pr(\mathbf{s}|\mathbf{y}), \tag{2.35}$$

where $Pr(\mathbf{s}|\mathbf{y})$ is the *'a posteriori'* probability that **s** is transmitted given that **y** is received. The ML-D boils down to selecting the signal among all candidate transmitted signals that minimises the below metric [99]

$$\hat{\mathbf{s}}^{MLD} = \min_{\mathbf{s} \in \mathbf{S}} \quad \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2, \tag{2.36}$$

where **S** is the set of possible transmit symbol vectors defined by the type of modulation scheme and the number of transmit antennas. The optimisation problem of (2.36) can be solved by a "brute-force" search over $\mathbf{S} = |\mathbb{M}|^{N_t}$, resulting in a prohibitive computational complexity that grows exponential with the number of decision variables (where $\mathbb{M}$ is the constellation set). This renders optimal detectors impractical in real MIMO systems.

2.6.2 Sub-Optimum Detectors

Several suboptimal detectors have been proposed to deal with the high computational complexity of the optimal detectors [2, 3, 100–102]. The suboptimal detectors are categorised into linear and non-linear, and they are briefly discussed below

2.6.2.1 Linear Detectors

In a linear signal detection scheme, all transmitted signals are considered interference except the desired data streams from the intended transmit antenna. Hence, interfering signals from other transmit antennas are suppressed when detecting the intended signal from the source transmit antenna. Typically, the design of linear detection schemes involves a linear linear transformation of the received symbol vector with the filtering matrix at the receiver to recover the transmitted symbol [2]. Mathematically, any linear detector can be expressed as

$$\hat{\mathbf{s}} = \mathcal{D}^{Lin} \cdot \mathbf{y}, \tag{2.37}$$

where $\mathcal{D}$ is the linear filtering or transformation matrix, which is the design criterion using various criteria.

1. **Match Filter Detector (MFD):** This is the simplest of all the linear detectors is a match filter detector, and it is also called maximum raio combining (MRC) because it maximizes the SNR of individual streams while ignoring

the mulitiuser interference. The estimated symbol from the received signal is given by [101]

$$\hat{\mathbf{s}}^{MFD} = \mathbf{H}^H\mathbf{H}\mathbf{s} + \mathbf{H}^H\mathbf{n}, \tag{2.38}$$

here, the linear transformation matrix $\mathcal{D}^{MFD} = \mathbf{H}^H$. An MFD is essentially based on the single-user detection philosophy; hence it does not belong to the category of joint MIMO detection schemes [2]. This explains why it has the least performance in MIMO systems compared to other linear detectors.

2. **Zero-Forcing Detector (ZF-D):** This belongs to the class of joint detection based MIMO detectors derived by inverting the channel matrix. Assuming zero noise power and a system with symmetrical channel $(N_t = K)$, such a system can be solved because there is an equal number of equations as there are unknowns and **H** is a square matrix of full rank. The equalisation/detector matrix is given by [100]

$$\mathcal{D}^{ZFD} = \left(\mathbf{H}^H\mathbf{H}\right)^{-1}\mathbf{H}^H, \tag{2.39}$$

$\mathbf{H}^{\dagger} = \left(\mathbf{H}^H\mathbf{H}\right)^{-1}\mathbf{H}^H$is the Moore-Penrose pseudoinverse of **H**, and is particularly important for an asymmetrical channel, where $K > N_t$ with a full column rank. The estimated symbol vector is obtained as

$$\hat{\mathbf{s}} = \mathcal{D}^{ZFD}\mathbf{y} = \mathbf{s} + \mathbf{n}\mathbf{H}^{\dagger}. \tag{2.40}$$

We can observe from (2.40) that the co-channel interference is completely cancelled, but the noise power is amplified.

3. **Minimum-Mean-Square-Error Detector (MMSE-D):** This detector is designed based on the MMSE criterion, which minimises the mean-square error between the original symbol vector and the channel's output symbol vector after applying the linear transformation matrix $\mathcal{D}^{MMSED}$, which is obtained

by solving the following optimisation problem [2]

$$\mathcal{D}^{MMSED} = \underset{\mathcal{D}^{MMSED}}{\arg\min} \quad \mathbb{E}\Big\{ ||\mathbf{s} - \mathbf{H}\mathbf{y}||^2 \Big\}. \tag{2.41}$$

From (2.41), we have

$$\mathcal{D}^{MMSED} = \left(\mathbf{H}^H\mathbf{H} + \alpha\mathbf{I}\right)^{-1}\mathbf{H}^H, \tag{2.42}$$

where $\mathbf{I} \in \mathbb{R}^{K\times K}$ identity matrix, α is the SNR for MMSE and when $\alpha = 0$, the problem reduces to ZF. The transmitted symbol in ZF is affected by the presence of coloured noise and leads to performance degradation. On the other hand, MMSE suppresses the noise enhancement as shown, but assumes knowledge of the noise variance. The estimated symbols from the output of the MMSE detector is thus

$$\hat{\mathbf{s}} = \mathcal{D}^{MMSED}\mathbf{y}. \tag{2.43}$$

2.6.2.2 Nonlinear Detectors

Nonlinear detectors are the ones that are defined by the nonlinear relationship between the equalisation or transformation matrix and the received output symbol vector at the receiver. We shall briefly explain the once relevant to this work below

Sphere Decoding (SD): The main idea behind the SD algorithm is to search only through the constellation points enclosed within a sphere with a specific, predetermined radius $\bar{d}$ [103], from the transmitted symbols. The maximum likelihood solution is obtained by searching and successfully marking all the constellation points or nodes within this radius. It is important to note that the SD's complexity is reduced by restricting the search within the sphere with a predetermined radius [2]. The estimated symbol from the received symbol at the receiver using SD is expressed as

$$\hat{\mathbf{s}} = \underset{\mathbf{s}\in\mathbb{C}^{N_t}}{\arg\min} \quad \Big\{ ||\mathbf{s} - \mathbf{H}\mathbf{y}||^2 \Big\} \leq \bar{d}^2. \tag{2.44}$$

SD was derived from the ML formulation to circumvent the exponential complexity of the ML detection [104]. However, its average complexity still grows exponentially with the number of decision variables (transmit antennas), and it is therefore not very hardware-friendly for m-MIMO applications [104].

2.6.3 Optimisation-based Detectors

Contrary to other MIMO detectors, these detectors are based on the semidefinite programming (SDP). Once the problem is convex, powerful numerical algorithms such as interior-point methods can efficiently solve it. Typical optimisation-based detectors are based on quadratically constrained quadratic programming (QCQP) to produce symbol detection at a lower computational cost than an ML-D. We express the ML-D optimisation problem of (2.36) as

$$\begin{aligned} &\min_{\bar{\mathbf{s}}_i \in \mathbb{R}^{2N_T \times 1}} && \left(\bar{\mathbf{H}}^T \bar{\mathbf{H}} \bar{\mathbf{s}} - 2\bar{\mathbf{s}}^T \bar{\mathbf{H}}^T \bar{\mathbf{y}} + \|\bar{\mathbf{y}}\|^2 \right) \\ &\text{s.t.} && \forall \mathbf{s}_i \in \{1, ..., 2N_t\} \end{aligned} \tag{2.45}$$

where, the symbol vector, channel matrix and the received symbol vector are defined in their equivalent real domains as follows

$$\bar{\mathbf{y}} \equiv \begin{bmatrix} \Re\{\mathbf{y}\} \\ \Im\{\mathbf{y}\} \end{bmatrix} \in \mathbb{R}^{2K \times 1}, \; \bar{\mathbf{s}} \equiv \begin{bmatrix} \Re\{\mathbf{s}\} \\ \Im\{\mathbf{s}\} \end{bmatrix} \in \mathbb{R}^{2N_t \times 1} \tag{2.46}$$

$$\bar{\mathbf{H}} \equiv \begin{bmatrix} \Re\{\mathbf{H}\} & -\Im\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} & \Re\{\mathbf{H}\} \end{bmatrix} \in \mathbb{R}^{2K \times 2N_t} \tag{2.47}$$

By defining new variables, we can write (2.45) as

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{trace}(\mathbf{LX}) \\ \text{s.t.} \quad & \text{diag}(\mathbf{X}) = \mathbf{I} \\ & \mathbf{X}(2N_t+1, 2N_t+1) = 1 \\ & \mathbf{X} \succeq 0 \\ & \text{rank}(\mathbf{X}) = 1 \end{aligned} \tag{2.48}$$

where: $\mathbf{L} = \begin{bmatrix} \bar{\mathbf{H}}^T\bar{\mathbf{H}} & -\bar{\mathbf{H}}^T\bar{\mathbf{y}} \\ -\bar{\mathbf{y}}^T\bar{\mathbf{H}} & ||\bar{\mathbf{y}}||^2 \end{bmatrix}$; $\mathbf{X} = \bar{\mathbf{s}}^T\bar{\mathbf{s}}$.

The difficulty of solving (2.48) lies with the rank-1 constraint (nonconvex constraint). This can be can be made convex via semidefinite relaxation (SDR), by removing the above rank constraints [105].

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{trace}(\mathbf{LX}) \\ \text{s.t.} \quad & \text{diag}(\mathbf{X}) = \mathbf{I} \\ & \mathbf{X}(2N_t+1, 2N_t+1) = 1 \\ & \mathbf{X} \succeq 0 \end{aligned} \tag{2.49}$$

The SDP based MIMO detectors have recently gained ample research attention [105–108]. The most attractive feature of the SDP-aided detectors is that they offer high performance in certain circumstances with worst-case computational complexity in polynomial-time.

2.6.4 Iteration-Based Detector

The most popular detector in this category is the one based on Approximate Message Passing (AMP) algorithm. An AMP is an iterative algorithm originally designed for signal regeneration in compressed sensing (CS) [109]. If we assume the MIMO model as a signal recovery problem, the compressed noisy measurement of the received vector is similar to (2.1). The iteration involved in an AMP algorithm is in two phase: the linear estimation (LE) based on (2.1) and symbol-by-symbol non-linear estimation (NLE), the problem is expressed as [109]

$$\textbf{LE:}\quad \mathbf{x}^t = \mathbf{s}^t + \mathbf{H}^H(\mathbf{y} - \mathbf{H}\mathbf{s}^t) + \mathbf{x}^t_{\text{Onsager}} \tag{2.50a}$$

$$\textbf{NLE:}\quad \hat{\mathbf{s}}^{t+1} = \eta_t(\mathbf{x}^t) \tag{2.50b}$$

where η_t denotes Lipschitz continuous function of $\mathbf{x}^t$ and $\mathbf{x}^t_{\text{Onsager}}$ is called the "Onsager term"(iterative thresholding), which regulates the correlation problem during t-th iterative process. The final estimate from the received signal vector is given by (2.50b).

In the context of signal detection, AMP uses an iterative thresholding to estimate the received signal by minimising the residual error in each successive t-th iteration, as expressed below [110]

$$\mathbf{x}^{t+1} = \mathbf{y} - \mathbf{H}\hat{\mathbf{s}}^t + \frac{K}{N_t} \cdot \frac{\sigma_s^2}{\sigma_s^2 + \alpha^t}\mathbf{x}^t;\ \forall t = \{0, \cdots, n\}, \tag{2.51}$$

$$\alpha^{t+1} = \sigma_n^2 + \frac{K}{N_t} \cdot \frac{\alpha^t \sigma_s^2}{\sigma_s^2 + \alpha^t}, \tag{2.52}$$

where σ_s^2 and σ_n^2 are transmit symbol and noise variances, respectively, and α^t is initialised with the initial signal estimate. Therefore, the final symbol estimate is

$$\hat{\mathbf{s}}^{t+1} = \frac{\sigma_s^2}{\sigma_s^2 + \alpha^t}\left(\mathbf{H}^H\mathbf{x}^{t+1} + \hat{\mathbf{s}}^t\right). \tag{2.53}$$

An APM algorithm has a unique ability to handle high-dimensional problems, hence it is a good candidate for massive MIMO detection [20]. Several variants of AMP-based detectors for MIMO detection have been reported in the literature [20, 110–112]. However, convergence is always the problem of an AMP algorithm, and therefore it requires a mechanism to facilitate its convergence. To deal with this problem, a damping mechanism was introduced to expedite the convergence of the AMP algorithm [113].

Furthermore, a modified AMP algorithm involving uncorrelated LE and a divergence-free NLE named orthogonal AMP (OAMP) was proposed in [114]. In this structure, the “Onsager term” is removed, causing the errors to be statistically orthogonal. Because of this, OAMP achieves a lower BER and faster convergence speed in many scenarios compared with AMP, especially for ill-conditioned channel matrices. Finally, due to low-cost iterative nature of an AMP algorithm, it can be easily unfolded and translated into sequence of DL layers used in MIMO receiver designs [23, 115, 116].

2.6.5 DL-based MIMO Detectors

These are MIMO detection schemes that use a DNN architecture. An example of such detectors that are most relevant to our proposed DL-based detector are DetNet [117] and OAMP-Net [23] because of their promising performances compared to others.

DetNet: This network is inspired by iterative gradient descent optimisation and performs well in, i.i.d complex Gaussian channel, achieving near-optimal performance with lower modulation schemes such as binary phase-shift keying (BPSK) and quadrature phase-shift keying (QPSK). However, DetNet architecture is very complex, with many layers and sub-layers. It also does not perform well with higher modulation schemes, such as 16-QAM, as we shall see in Chapter 6, subsection 6.8.4. The architecture is based on the formulations derived from the maximum likelihood expression as follows:

$$\mathbf{x}_{r+1} = \hat{\mathbf{s}}_r - \Psi^{[1]}_{r+1}\mathbf{H}^H\mathbf{y} + \Psi^{[2]}_{r+1}\mathbf{H}^H\mathbf{H}\hat{\mathbf{s}}_r \tag{2.54a}$$

$$\mathbf{u}_{r+1} = \left[\tilde{\Theta}^{[3]}_{r+1}\mathbf{x}_{r+1} + \tilde{\Theta}^{[4]}_{r+1}\mathbf{a}_{r-1} + \theta^{[5]}_{r+1}\right]_+ \tag{2.54b}$$

$$\mathbf{a}_{r+1} = \tilde{\Theta}^{[6]}_{r+1}\mathbf{u}_{r+1} + \theta^{[7]}_{r+1} \tag{2.54c}$$

$$\hat{\mathbf{s}}_{r+1} = \tilde{\Theta}^{[8]}_{r+1}\mathbf{u}_{r+1} + \theta^{[9]}_{r+1}, \tag{2.54d}$$

here, Ψ, θ and $\tilde{\Theta}$ represent the set of training parameters, $[z]_+ = \max(z, 0)$, $\mathbf{a}$ and $\hat{\mathbf{s}}$ are the auxiliary and recovered transmitted symbols of the r-th layer iterations.

The unique property of such detectors is their ability to sustain their performance under higher-dimensional signals [117]. More recent works have involved

MIMO detection through DL. One of the earliest attempts is the work of O'Shea *et al.* [118], who implemented unsupervised learning using an auto-encoder as an extension of end-to-end learning of previous attempts [119]. Channel equalisation for the nonlinear channel using a DNN was proposed by Xu *et al.* [120], where two neural networks are jointly trained. The first is a CNN, which is trained to recover the transmitted symbols from nonlinear distortions and channel impairments. The second is an MLP, also known as the fully connected neural network, and is used to perform the detection.

The growing popularity of unfolding iterative optimisation algorithms through projected gradient descent (deep-unfolding) to design DNN to solve a spectrum of applications has led to a paradigm shift for efficient learning-based solutions for the physical layer design [121]. One of the successful applications of deep-unfolding for MIMO detection is the "DetNet" proposed by Samuel *et al.* [22]. The approach is significant as it derives a learnable signal detection architecture for multiple channels on a single training shot with near-optimal performance and also works well under both constant and Rayleigh fading channels. Multilevel MIMO detection using coupled-neural networks structure is investigated by Corlay *et al.* [122]. The network uses a multi-stage sigmoid activation function and a random forest approach to reduce the detection complexity with relatively fewer parameters. A similar approach by unfolding belief propagation (BP) based on modified BP algorithms (damped BP and maximum BP) is later introduced by Tan *et al.* [123] and Liu and Li [124]. The work proposed in [22] has been further extended to handle higher digital constellations [117], where the authors investigate the complexity-accuracy trade-off as more layers are added.

OAMP-Net: Beyond using projected gradient descent approaches with DNN architectures, other lower-cost learning-based detectors based on iterative AMP algorithms are: "trainable iterative detector (TI-detector)" proposed by Imanishi *et al.* [125], "orthogonal approximate massage passing deep network (OAMP-Net)" introduced by He *et al.* [23] and "trainable projected gradient detector (TPG-Net)" proposed by Takabe *et al.* [126]. The OAMP-Net architecture is expressed as

$$\mathbf{x}_{r+1} = \hat{\mathbf{s}}_r + \Psi_{r+1}^{[1]} \mathbf{H}^H \left(\vartheta \mathbf{H}^H \mathbf{H} + \sigma_n^2 \mathbf{I} \right)^{-1} \left(\mathbf{y} - \mathbf{H} \hat{\mathbf{s}}_r \right) \tag{2.55a}$$

$$\hat{\mathbf{s}}_{r+1} = \eta_r \left(\mathbf{x}_r ;\ \sigma_r^2 \right), \tag{2.55b}$$

where Ψ, ϑ are the trainable parameters and η is the denoiser or the same "Onsager term" used by AMP and σ_r^2 is the prior variance that influences the accuracy of $\hat{\mathbf{s}}_{r+1}$ and is described in [23].

Contrary to previous learning-based detectors that heavily depend on a huge amount of parameters, these models exploit full domain knowledge to achieve acceptable performance with fewer parameters. However, these algorithms require channel inversion at every training and inference steps to compute the nonlinear estimator for symbol detection.

2.7 Summary

The chapter presents the general overview of MIMO system and its applications in physical layer communications. We have concisely reviewed the existing traditional precoding and detection schemes. Transmission preprocessing techniques are generally divided into closed-form precoding and optimisation-based precoding approaches. The closed-form precoding can be categorised into linear and non-linear schemes. The linear precoding schemes are designed based on the channel's knowledge to cancel the MUI with a relatively low computational cost. The non-linear precoding schemes can achieve added performance gains over linear methods at the expense of increased computational complexities. In the precoding designs, we specifically focus on the conventional precoding schemes, optimisation-based, and CI-based downlink precoding methods for power minimisation problems. The post-processing downlink transmission at the receiver or user-end requires decoding the transmitted data symbols from the received symbols with minimum error probability. As in precoding designs, we have briefly presented the review of linear, non-linear, iteration and optimisation based detection methods that are relevant to this work.

Chapter 3

Machine Learning for 5G Communications and Beyond

This chapter introduces the basic concepts of machine learning and its applications in wireless physical layer communications. In particular, we present an overview of DL techniques and their potential benefits in designing end-to-end learning-based communication systems. The ML literature is vast, and we only review the relevant ML techniques for conciseness. Furthermore, we present a brief review of the required theoretical concepts leading to the generic, scalable deep neural network designs for MIMO detection and multi-user precoding schemes.

3.1 Types of Machine Learning

Depending on the specific applications, ML can be divided into four broad learning schemes as summarised in Table 3.1. Unlike in computer vision and other domains, where labelled training data is readily available, such is challenging to obtain in

Table 3.1: Taxonomy of Machine Learning

Machine Learning Types							
Supervised Learning		Unsupervised Learning		Semi-Supervised Learning		Reinforcement Learning	
Classification	Regression	Clustering	Regression	Classification	Regressions	Classification	Control
Support Vector Machine	Linear Regression	K-Means Algorithm	Linear Regression	Support Vector Machine	K-Means Algorithm	Support Vector Machine	Deep Q-Learning
Naive Bayes	Ensemble Schemes	K-Medoids	Ensemble Schemes	Naive Bayes	K-Medoids	Naive Bayes	Model-Based Reinforcement Learning
Discriminant Analysis	Decision Trees	C-Means, Fuzzy	Decision Trees	Discriminant Analysis	C-Means, Fuzzy	Discriminant Analysis	Markov Decision Model
Nearest Neighbour (K-Nearest)	Scalable Vector Graphics	Hidden Markov Model	Deep Auto-encoder	Nearest Neighbour (K-Nearest)	Hidden Markov Model	Deep Neural Networks	Q-Learning
Deep Neural Networks	Deep Neural Networks	Hierarchical	Deep Unfolding	Deep Neural Network	Deep Auto-encoder		
		Gaussian Mixture					
		Principal Component Analysis					
		Deep Neural Networks					

wireless communication. Therefore, Chapter 4 and Chapter 5 of this thesis focus on an unsupervised learning scheme.

3.1.1 Unsupervised Learning

Unsupervised learning is the type of learning that is independent of the pre-defined examples. The index or category associated with each input data is not given. In other words, the input examples are not labelled (unstructured dataset) [127]. In Figure 3.1, the input data is unstructured (unlabelled). The main task of the algorithm is to find out the structure in the data by organising and grouping the data points with similar characteristics into the same cluster or group. This algorithm is much more sophisticated than supervised learning and mimics how the human brain processes data. Most often than not, in classification problems, the data is labelled. However, in reality, we do not have the luxury of having labelled data because it is challenging to annotate. This unique property of unsupervised learning could potentially be suitable for end-to-end learning over the air, wireless channel estimation, and determining users' nature, behaviours, and mobility in a cognitive radio environment [129]. Contrary to supervised learning, where a Cross-Entropy or Minimum-Mean-Squared-Error is a loss function to compare the output of the prediction with the ground truth, we use the original objective function as a loss function in this thesis.

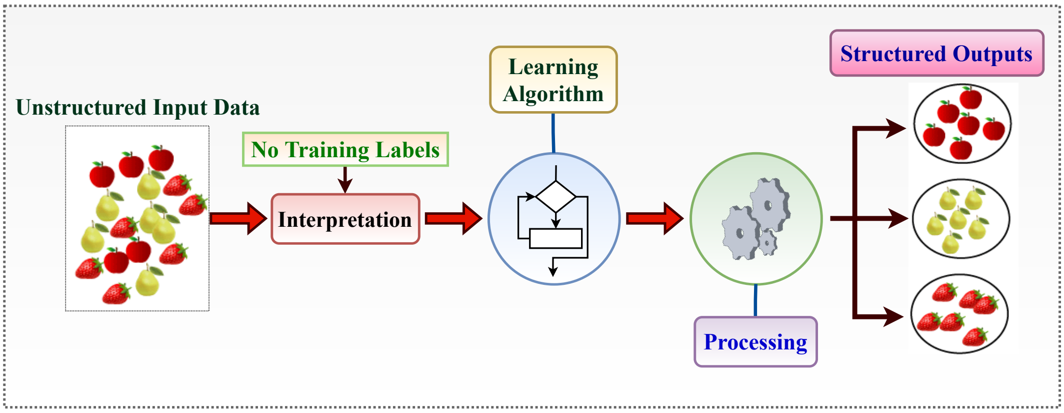


Figure 3.1: Unsupervised Learning Work-flow [128]

3.2 Deep Learning

Without loss of generality, simple ML algorithms used with a relatively small amount of training dataset do not generally perform well when a large amount of data is used [130]. For this reason, therefore, more robust ML technique is necessary. Classical ML techniques, such as: support vector machine (SVM), nearest neighbour, decision trees, principal component analysis (PCA), k-means clustering, etc cannot process large amount of data [130]. Their accuracy declines with the increase in the dimension of the datasets, albeit proven very ineffective in solving many problems [131]. This leads to developing DL techniques that use a set of algorithms to exploit high-level abstraction in the data. The evolution of DL ensued concurrently with the study of AI, particularly the study of artificial neural networks (ANN) in the 1980s [131].

3.2.1 Artificial Neural Network (ANN)

An ANN is a processing unit inspired and designed based on the natural architecture and function of the human or animal brain [127]. Typically, ANN is represented by a single unit (layer) and a firing node called activation function through which some nonlinearity is applied to make it able to learn. As seen in Figure 3.2, the learning is accomplished by adjusting the connection between the inputs $\mathbf{x} = [x_1, \cdots, x_n]$ and weights $\mathbf{w} = [w_1, \cdots, w_n]$ to produce prediction at the output nodes.

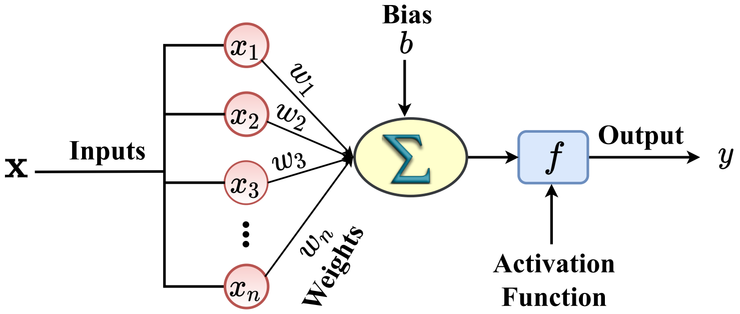


Figure 3.2: Simple Artificial Neural Network [127]

Deep neural network (DNN): This is a more sophisticated form of ANN with many processing units (neurons/nodes) and layers. A DNN is a nonlinear ML model that provides a fair, accurate universal approximation of any function. It requires no prior assumption of the model to build the process [131]. Figure 3.3 shows a typical DNN architecture composed of three basic components: an input layer, a hidden layer, and output layer. The number of nodes (analogous to connecting neurons in the human brain) depends on the number of input and output variables (parameters). In between the first and the last layers are hidden layers, whose number is arbitrarily chosen depending on the size of the input data available and the task at hand. Information is transmitted between layers through the connecting nodes (neurons) with the aid of the connecting weights. Each layer is associated with weights and biases parameterised by $\bar{\theta}$. The network acquires intelligence by adjusting the values of these weights and biases through a repetitive training (learning algorithm). A fully feed-forward DNN as the one shown in Figure 3.3 is sometimes called a multilayer perceptron (MLP).

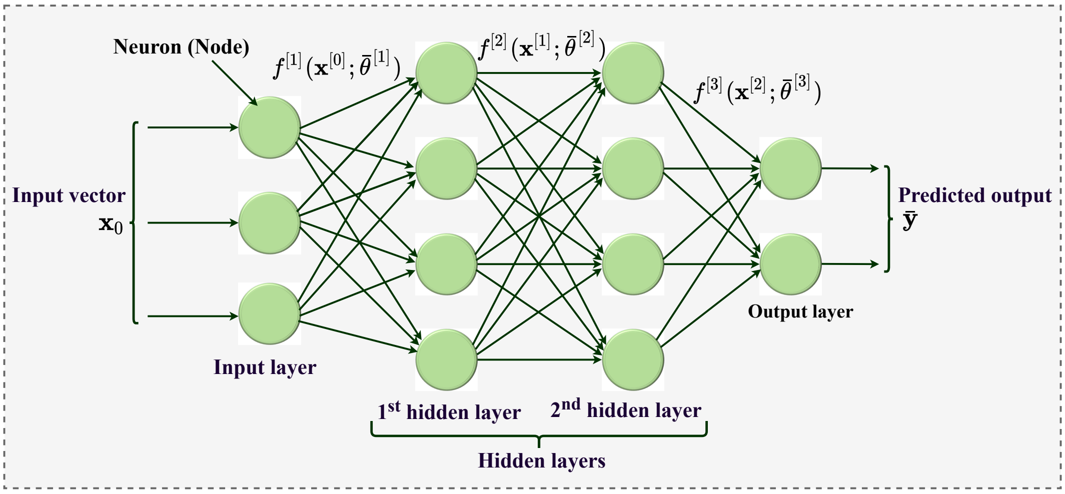


Figure 3.3: A typical three layer fully connected feed-forward DNN

Suppose the network is fed with an $\mathbf{x}_0$ m-th size input vector and produces an n-th size output vector $\bar{\mathbf{y}} \in \mathbb{R}^n$. By changing the parameter vector $\bar{\theta} \in \mathbb{R}^k$ of k-th dimension, we can model different input-output relations that share the same basic structure determined by the original choice of the function $f(\cdot)$. The output from each layer is deterministically determined by the function $f(\cdot)$ as follows

$$\mathbf{x}^{[l]} = f^{[l]}(\mathbf{x}^{[l-1]};\bar{\theta}^{[l]}), \ \forall l = \{1,\cdots,L\} \tag{3.1}$$

where L is the number of layers, including the input layer that determine how deep the DNN is. Accordingly, the feed-forward is defined by the composite function as [127]

$$f^{[l]}(\mathbf{x}^{[l-1]};\bar{\theta}^{[l]}) = \sigma\left(\mathbf{W}^{[l]}\mathbf{x}^{[l-1]} + \mathbf{b}^{[l]}\right) \tag{3.2}$$

where σ is the nonlinear activation function, $\mathbf{W}^{[l]}$ and $\mathbf{b}^{[l]}$ represent the weights and biases, respectively. Finally, the overall input-output relation is thus expressed as

$$\bar{\mathbf{y}} = f^{[L-1]}\left(f^{[l-2]}\left(\mathbf{x}^{[L-3]};\bar{\theta}^{[L-2]}\right) \cdots f^{[1]}\left(\mathbf{x}^{[0]};\bar{\theta}^{[1]}\right);\bar{\theta}^{[L-1]}\right) \tag{3.3}$$

Convolutional neural network (CNN): The CNN is very similar to typical MLP because they both have learnable weights and biases based on the same theoretical foundation [132]. As earlier explained, in a typical MLP, a neuron in the input layer is connected to the neuron in the output layer. However, only a small area of the input layer neuron is connected to the hidden layer's neuron in CNN. These areas or regions are called local receptive fields [127]. Unlike MLP, where neurons in a single layer function are entirely independent and do not share any connections, the weights and biases are the same for all hidden neurons in a given layer in CNN. Furthermore, additional transformation known as pooling is applied after the activation to reduce the feature map's dimension into a single output. These unique properties, such as weight sharing and pooling, make the CNN network more efficient, significantly reducing the number of parameters than in a fully connected DNN (i.e. MLP).

Generally, convolution layer consists of $\bar{K}$ filters of size $f_w \times f_h$ that perform convolution operations as it scans through an input $I^l_{(W_{\text{in}} \times H_{\text{in}} \times C_{\text{in}})}$ tensor[1], a stride S, which denotes the number of pixels by which the window moves after each operation [128]. The resulting operations produce an output Y^l called a feature map or

[1] I^l is usually an image in computer vision; W_{in}, H_{in} and C_{in} are the width, height and the number of channels, respectively

activation map according to the following.

$$Y^{l}_{p,q} = \sum_{a=0}^{f_w-1} \sum_{b=0}^{f_h-1} \mathbf{W}_{(a,\, b)} I^{l-1}_{(p+a,\, q+b)}, \tag{3.4}$$

where $\mathbf{W}_{(a,\, b)}$ is the filter weight matrix.

3.3 Learning to Communicate

Intuitively, a communication system from transmitter to receiver is divided into several signal processing blocks, such as modulation, source coding, channel coding, channel estimation, and signal detection, in order to achieve a near theoretically optimal solution. The advantage of this configuration is that each component can be optimised separately, resulting in efficient and stable communication systems that are currently available, albeit being sub-optimal [133]. However, the concept of DL traces back to the original problem and tries to jointly optimise the transmitter and receiver without introducing the block configuration [118]. As an illustration, we consider a simple communications system, as depicted in Figure 3.4. The transmitter sends message $s = \{1, \cdots, \mathbb{M}\}$ over the channel to the receiver. The receiver generates the estimate $\hat{\mathbf{s}}$ of the original message.

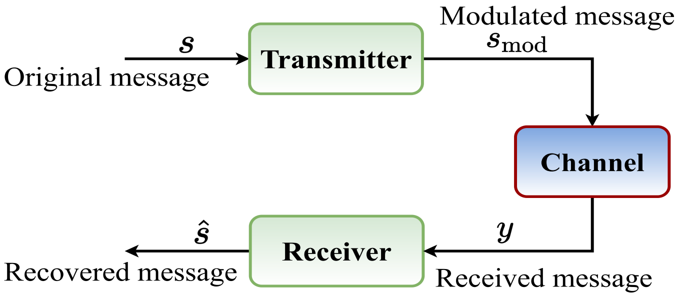


Figure 3.4: A simple communications system

DL has been applied in wireless physical layer communications because of its propensity to create an ingenious framework that can intelligently make decisions with some high-level degree of accuracy and a reasonably low online computational complexity. [24, 25, 118, 129, 134–138]. Therefore, the communications system in Figure 3.4 can be represented by an auto-encoder[2], replacing the transmitter and

[2] An auto-encoder is a DNN trained to reconstruct input data at the output.

receiver [139]. Generally, there are two primary paradigms for solving problems in ML: the data-driven and model-driven approaches.

3.3.1 Data-Driven DL

The data-driven DL method is the most commonly popular method applied in several computer vision problems, naturally due to the nonavailability of well defined tractable mathematical models [140, 141]. The data-driven approach relies on training labelled data to build a system that can identify the correct answer based on what the system has seen before. In this context, the models learn by parameter updates and hyperparameter tuning during training. There are several ways of doing this, but the most popular is using NN algorithms in different forms. The essential ingredients for this method is the availability of suitable labelled dataset. It is important to note that this approach does not require humans to describe a set of rules accurately. With data-driven DL, the system learns on its own to make accurate predictions when presented with the new dataset based on the training data it had initially seen. Depending on the application, the larger the training data, the better the system can be.

While data-driven DL has been widely and successfully applied in different domains, it may not be suitable in some fields that do not have the luxury of labelled training dataset, especially wireless communications. Moreover, the relationship between the NN and the network topology has limited theoretical foundations, making the structure vague and unpredictable [142]. These two reasons limit the widespread adoption of data-driven DL for physical layer communications applications. To deal with this issue, model-driven DL that exploits the expert's knowledge based on the physical mechanism of the system has been proposed [11, 142].

3.3.2 Model-Driven DL

The model-driven DL approach is based on a deep understanding of the system and the relationships between its constituent components or variables from the physical system. This approach uses physical mechanism and domain knowledge to build the learning architecture and therefore does not require a large amount of training data, making it is easier to train [142]. The model-driven DL has three main components: model, algorithm and network. Unlike the traditional analytic model-driven approach, model-driven DL method provides only a rough and broad description of the solution, reducing the demand for an accurate modelling strategy. The algorithm is then unfolded into learning layers to form the network, which is trained via backpropagation. Model-driven DL methods are appealing for solving physical layer communication problems because of the availability of mathematical models and less demand for a large amount of training data [11].

The physical layer communications system is a well-researched field based on established theoretical foundation with well-defined problems. However, most existing algorithms for solving such problems are computationally intractable for implementation on practical systems [11]. A model-driven DL can compensate for the imperfection and inaccuracies arising from modelling by learning a set of parameters of the unfolded learning network. Because the model-driven DL model can be efficiently trained with small training data, it has a relatively short training time and is less prone to overfitting [142]. A logical question one may ask is whether we can design learning architecture based on theoretical foundations and make the network explicable and predictable. We answer this question by using a model-driven DL derived from the analytical model and associated algorithm.

The model-driven DL is based on the physical mechanism and domain knowledge for a specific problem. This is contrary to the data-driven DL approach that uses a standard NN architecture as a black box and massive data to train the NN. It is important to note that the pure model-driven approach can provide an optimal solution when the model is sufficiently accurate with the optimisation algorithm being deterministic. A fatal defect of the pure model-driven method rests in the inability to model a specific task in real applications accurately. On many occasions, obtaining an accurate model is often challenging. The integration of learning architecture with model-driven methods has recently received much attention and is becoming a potential strategy for designing intelligent communication systems with several promising results [11, 12, 23, 143].

3.4 Scalable DNN for Inference Acceleration

The accomplishments of DNN can be mainly attributed to the ever-increasing computing power and availability of data. As the parallel powerful, fast computing nodes, such graphical user unit (GPU) and field-programmable gate array (FPGA) become available at a lower cost, we can use massive data to train DNN with more layers and neurons, resulting in higher inference accuracy. For many applications, the sizes of the network have grown drastically overtime beyond the petascale. The number of popular deep neural networks' parameters has considerably increased for many application domains, particularly in computer vision, autonomous driving, to mention a few over the last twenty years [144]. Such large models may not offer significant hurdles for applications where powerful computing resources are easily accessible through network connections. However, running trained networks on embedded hardware platforms, where security, privacy and latency are of significant considerations, the inference must be done locally or at the network's edge. Such computation is subject to severe constraints (memory and power) due to the limited available resources. For this reason, there has been a recent drive to reduce the DNN model size, driven from the image processing research [145].

Traditionally, DNN is designed with full-precision weights and activations. This can result in significant memory consumption and computational complexity. The DNN complexity reduction and acceleration techniques can be broadly classified into three categories:

i. Structured simplification: This involves a systematic approach of network factorisation (factorises a convolutional layer into many efficient ones), channel pruning, sparse connections to reduce the size of the DNN model [146].

ii. Quantization: In this technique, the computations involving weights, activations, and sometimes input tensors are performed at lower bit-widths than floating-point precision [145].

iii. Optimised Implementation: This approach uses Fast Fourier Transform (FFT) based on NVIDIA's cuFFT library to provide significant speedups [147].

Among the above three model simplification techniques, the first two are most appealing because:

- They provide good insight to understanding the internal dynamics of the hidden layer operations, which is usually impossible with the conventional DNN.
- They both lead to model size reduction, improve training and inference efficiencies and reduces hardware requirements during model deployment on the edged-devices.
- Specifically, for a quantised network, most multiply accumulates (MACs) operations required to compute the neurons' weighted sums are replaced by simple binary operations (bit-wise or XNOR operations).

For these reasons mentioned above, we propose stochastic quantisation based on standard binary and ternary quantisation in Chapter 4 and a novel structural DNN simplification we shall describe in Chapter 5.

3.4.1 NN Weight Pruning

Neural networks pruning is an old concept as far back as 1990 and before. Some trainable parameters of NN are redundant and do not contribute much to the learning process [148]. Not all neurons contribute to the output (learning process); some are redundant. In simple term, pruning is a technique that involves removing parameters, usually layer neurons, from the neural network so that the network's accuracy is maintained while improving its efficiency. Typically, pruning is divided into structured and unstructured depending on the designer's preference between speed and memory efficiency [148]. Structured pruning entails carefully removing a substantial part of the network, such as a layer or a channel. Technically speaking, structured pruning prunes weights in groups. On the other hand, unstructured pruning requires finding and eliminating the less significant connection wherever they are in the network. The unstructured pruning does not consider any relationship between the pruned weights.

3.5 Summary

This chapter presents a detailed overview of ML algorithms that are pertinent to the thesis. Beyond the traditional applications of ML, particularly DL in computer vision, the chapter explains the conceptual DL frameworks for designing signal processing techniques for physical layer communications. The advantage of adopting DL for physical layer communications designs is highlighted. The chapter also explains how data-driven and model-driven DL methods are combined to enhance the performance of the learning-based signal processing sachems. We have demonstrated how conventional iterative algorithms are converted into learning iteration learning layers. Finally, this chapter presents a synopsis of classical NN model compression, leading to more flexible learning frameworks.

Chapter 4

Unsupervised Deep Unfolding for Symbol Level Precoding Design

The chapter extends the traditional precoding design concept explained in Chapter 2, to a learning-based approach. Specifically, we focus on building an unsupervised learning framework via model-driven DL methods. Accordingly, we design learning-based precoding schemes that exploit known interference in MU-MISO systems for the power minimisation problem under SINR constraints. The learning framework is designed by unfolding an IPM iterative algorithm via *'log'* barrier function derived from SLP power minimisation formulation. The proposed learning-based precoding scheme does not require generating the training dataset from the conventional optimisation solutions, thereby saving considerable computational effort and time.

4.1 Introduction

Traditionally, interference is regarded as the limiting factor against the ever-increasing needs for transmission rates and QoS in 5G wireless communication systems and beyond [49, 149, 150]. However, recent studies on interference exploitation have transformed the traditional paradigm in which known inferences are effectively managed [49, 149–152]. Consequently, transmit beamforming techniques for the downlink channels for power minimisation problems under specific QoS become imperative for high-throughput systems under interference.

The idea of exploiting interference was first introduced by Masouros and Alsusa [153], where instantaneous interference was classified into constructive and destructive. Initial sub-optimal approaches to exploit CI were first introduced by Masouros *et al.* [61, 154]. The first form of optimisation-based CI precoding was introduced in the context of vector perturbation precoding through a quadratic optimisation approach [60]. A convex optimisation-based CI scheme termed symbol-level-precoding technique was proposed first with strict phase constraints on the received constellation point [65], and with a robust relaxed-angle formulation [49]. As a result of the performance gains over conventional BLP schemes, the idea of CI has been applied in many domains, such as vector perturbation [155], wireless information and power transfer [156], mutual coupling exploitation [82], multi-user MISO downlink channel [83], directional modulation [157], relay and cognitive radio [149, 158]. Despite the superior performance offered by CI-based precoding methods, their increased computational complexity can hinder their practical application when performed on a symbol-by-symbol basis. To address this, Li and Masouros [21] proposed an iterative closed-form precoding design with optimal performance for CI exploitation in the MISO downlink by driving the optimal precoder's mathematical Lagrangian expression and Karush–Kuhn–Tucker conditions for optimisation with both strict and relaxed phase rotations.

Contrary to the conventional way of training DNN in a supervised learning fashion without the specifics of the objective functions as explained in the Chapter 2, in this chapter, we adopt an unsupervised learning approach. Specifically, we focus on the power minimisation problem via SLP and show the low computational cost of our proposal over the traditional optimisation SLP schemes. The contributions of this chapter are summarised below:

- We introduce an unsupervised DNN-based power minimisation SLP scheme for MU-MISO downlink transmission. The proposed framework is designed by unfolding an IPM algorithm via a *'log'* barrier function that exploits the convexity associated with the SLP inequality constraints. The learning framework utilises the domain knowledge to derive the Lagrange function of the original SLP optimisation as a loss function. This is used to train the network in an unsupervised mode to learn a set of Lagrangian multipliers that directly minimise the objective function to satisfy the constraints. A regularisation parameter is added to the Lagrange function to aid the training convergence, and we provide detailed formulations leading to the unfolded unsupervised learning architecture for constrained optimisation problems.

- We extend the formulation to design a robust learning-based precoder where the uncertainty in channel estimation is considered.
- We derive analytic expressions for the computational complexity of various SLP and the proposed unsupervised learning precoding schemes. Our analysis demonstrates that the proposed deep unfolding (DU) framework offers a theoretical, computational complexity reduction from $\mathscr{O}(n^{7.5})$ to $\mathscr{O}(n^3)$ for the symmetrical system case where n = number of transmit antennas = number of users. This is reflected in a commensurate decrease in the execution time as compared to the SLP optimisation-based method.

4.2 System Formulation

Consider a single-cell downlink channel with N_t transmit antennas at the BS transmitting to K single-antenna users. Assume a quasi-static flat-fading channel between the BS and the users, denoted by $\mathbf{h}_i \in \mathbb{C}^{N_t \times 1}$. The received signal at user i is given by

$$\begin{aligned} y_i &= \mathbf{h}_i^H \sum_{k=1}^{K} \mathbf{w}_k s_k + n_i \\ &= \mathbf{h}_i^H \sum_{k=1}^{K} \mathbf{w}_k e^{j(\varphi_k - \varphi_i)} s_i + n_i \end{aligned} \tag{4.1}$$

where $\sum_{k=1}^{K} \mathbf{w}_k s_k = \sum_{k=1}^{K} \mathbf{w}_k e^{j(\varphi_k - \varphi_i)} s_i$ is the transmit signal and $s_i = s e^{j\varphi_i}$ is assumed to be a referenced PSK modulated symbol with constant amplitude s. Also, $\mathbf{h}_i$, $\mathbf{w}_i$, s_i, n_i and φ_i represent the channel vector, precoding vector, data symbol, received noise and phase rotation for the i-th user.

4.3 Proposed Learning-Based SLP for Power Minimisation

This section presents the formulation of a learning-based CI power minimisation problem for SLP. Throughout this section, we assume a perfect CSI known at the BS. Motivated by the recent adoption of an IPM for image restoration [159], we propose an unsupervised learning framework that unfolds a constrained optimisation problem into a sequence of learning layers/iterations for a multi-user MISO beamforming. We first convert CI-based optimisation problem of (2.31) defined in subsection **2.5.2** of Chapter **2** into a standard IPM formulation containing a slack variable, where necessary. The measure of the fidelity of the solution to (2.31) is determined by learning a set of penalty parameters in the form of Lagrange multipliers associated with the constraints. From the original SLP power minimisation problem (2.31), we define the following

$$\hat{\mathbf{h}}_i = \mathbf{h}_i e^{j(\varphi_1 - \varphi_i)}, \tag{4.2}$$

$$\mathbf{w} = \sum_{k=1}^{K} \mathbf{w}_k e^{j(\varphi_k - \varphi_1)}. \tag{4.3}$$

Accordingly, to ease the analysis, we partition the complex rotations into the real and imaginary parts as follows

$$\hat{\mathbf{h}}_i = \hat{\mathbf{h}}_{Ri} + j\hat{\mathbf{h}}_{Ii} \tag{4.4a}$$

$$\mathbf{w} = \mathbf{w}_R + j\mathbf{w}_I \tag{4.4b}$$

where $\hat{\mathbf{h}}_{Ri} = \Re(\hat{\mathbf{h}}_i)$, $\hat{\mathbf{h}}_{Ii} = \Im(\hat{\mathbf{h}}_i)$, $\mathbf{w}_R = \Re(\mathbf{w})$ and $\mathbf{w}_I = \Im(\mathbf{w})$. The product of complex rotations of (4.4a) and (4.4b) can be written as

$$\hat{\mathbf{h}}_i \mathbf{w} = (\hat{\mathbf{h}}_{Ri} + j\hat{\mathbf{h}}_{Ii})(\mathbf{w}_R + j\mathbf{w}_I). \tag{4.5}$$

The real and imaginary parts of (4.5) can be written in vector forms as follows

$$\Re(\hat{\mathbf{h}}_i\mathbf{w}) = \begin{bmatrix} \hat{\mathbf{h}}_{Ri} & \hat{\mathbf{h}}_{Ii} \end{bmatrix} \begin{bmatrix} \mathbf{w}_R \\ -\mathbf{w}_I \end{bmatrix} \tag{4.6a}$$

$$\Im(\hat{\mathbf{h}}_i\mathbf{w}) = \begin{bmatrix} \hat{\mathbf{h}}_{Ri} & \hat{\mathbf{h}}_{Ii} \end{bmatrix} \begin{bmatrix} \mathbf{w}_I \\ \mathbf{w}_R \end{bmatrix} \tag{4.6b}$$

Let $\Lambda_i = [\hat{\mathbf{h}}_{Ri} \;\; \hat{\mathbf{h}}_{Ii}]^T$, $\mathbf{w}_1 = [\mathbf{w}_R \; -\mathbf{w}_I]^T$ and $\mathbf{w}_2 = [\mathbf{w}_I \;\; \mathbf{w}_R]^T$

$$\Re(\hat{\mathbf{h}}_i^T\mathbf{w}) = \Lambda_i^T\mathbf{w}_1 \text{ and } \Im(\hat{\mathbf{h}}_i^T\mathbf{w}) = \Lambda_i^T\Pi\mathbf{w}_1 \tag{4.7}$$

where

$$\mathbf{w}_2 = \Pi\mathbf{w}_1 \text{ and } \Pi = \begin{bmatrix} \mathbf{O}_{N_t} & -\mathbf{I}_{N_t} \\ \mathbf{I}_{N_t} & \mathbf{O}_{N_t} \end{bmatrix};\ \in \mathbb{R}^{2N_t\times 2N_t} \tag{4.8}$$

Note that $\mathbf{I}_{N_t}$ is the identity matrix and $\mathbf{O}_{N_t}$ the matrix of zeros, respectively. Using the above definitions, problem (2.31) can be recast into its mutlicast formulation [49]

$$\begin{aligned} \min_{\{\mathbf{w}_1\}} \quad & \|\mathbf{w}_1\|_2^2 \\ \text{s.t.} \quad & \left|\Lambda_i^T\Pi\mathbf{w}_1\right| \leq \left(\Lambda_i^T\mathbf{w}_1 - \sqrt{\Gamma_i n_0}\right)\tan\phi \ , \ \forall i \end{aligned} \tag{4.9}$$

4.3.1 Interior Point Method

Consider a general form of a nonlinear constrained optimisation of the form [160]

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^{\mathbf{N}}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) \geq 0 \\ & C(\mathbf{x}) = 0\,. \end{aligned} \tag{4.10}$$

The rationale of adopting IPM is to substitute the initial constrained optimisation problem by a chain of unconstrained sub-problems of the form

$$\min_{\mathbf{x}\in\mathbb{R}^{\mathbf{N}}} f(\mathbf{x}) + \lambda C(\mathbf{x}) + \mu B(\mathbf{x}). \tag{4.11}$$

where $B(\cdot) \triangleq -\sum \ln(\cdot)$ is the logarithmic barrier function associated with inequality constraint with unbounded derivative at the boundary of the feasible domain, $C(\cdot)$ is a function associated with equality constraint, μ and λ are the Lagrangian multipliers for inequality and equality constraints, respectively. For K users, we define a vector $\mu \triangleq [\mu_1, \cdots, \mu_K]$.

Following the above line of argument, the unconstrained sequence of (4.9) per user can be written as

$$\min_{\mathbf{w} \in \mathbb{R}^{\mathbf{2N_t \times 1}}} f(\mathbf{w}_1) + \mu B(\mathbf{w}_1), \tag{4.12}$$

To facilitate the solution of (4.9) , we introduce additional notations. For every inequality constraint, $\gamma \in \{0, +\infty\}$ and $\mathbf{w}_1 \in \mathbb{R}^{2N_t \times 1}$, we define the proximity function as in [160] with respect to (4.12), which we shall later use to compute the projected gradient descent as

$$\text{prox}_{\gamma\mu B}(\mathbf{w}_0) = \underset{\mathbf{w_1} \in \mathbb{R}^{\mathbf{2N_t \times 1}}}{\text{argmin}} \quad \frac{1}{2}\|\mathbf{w}_0 - \mathbf{w}_1\|_2^2 + \gamma\mu B(\mathbf{w}_1), \tag{4.13}$$

where γ is the step-size for computing the gradients and $\mathbf{w}_0$ is the initial value of the precoding vector. To convert (3) into its equivalent barrier function problem, we integrate the inequality constraint into the objective by translating it into a barrier term as follows [161]

$$\begin{aligned} \min_{\mathbf{w_1}} \quad & f(\mathbf{w}_1) - \mu \sum_{i=1}^{p} \ln(g_i(\mathbf{w}_1)) \\ \text{s.t.} \quad & C(\mathbf{w}_1) = 0 \end{aligned} \tag{4.14}$$

where $g(\mathbf{w}_1) = \left(\Lambda_i^T \mathbf{w}_1 - \sqrt{\Gamma_i n_0}\right) \tan\phi - \left|\Lambda_i^T \Pi \mathbf{w}_1\right|$ and p is the number of inequality constraints.

Going back to our initial SPL optimisation to apply this framework, first we rewrite the constraint of (4.9) as

$$a \leq \Lambda_i^T \Pi \mathbf{w}_1 \leq b, \tag{4.15}$$

where

$$a = -\left(\Lambda_i^T \Pi \mathbf{w}_1 - \sqrt{\Gamma_i n_0}\right) \tan\phi, \tag{4.16a}$$

$$b = \left(\Lambda_i^T \Pi \mathbf{w}_1 - \sqrt{\Gamma_i n_0}\right) \tan\phi. \tag{4.16b}$$

Therefore, the original problem (4.9) becomes

$$\begin{aligned} \min_{\{\mathbf{w_1}\}} \quad & \|\mathbf{w}_1\|_2^2 \\ \text{s.t.} \quad & a \leq \Lambda_i^T \Pi \mathbf{w}_1 \leq b \ , \ \forall i. \end{aligned} \tag{4.17}$$

To obtain each user's precoder from (4.9), suppose the optimal solution to (4.9) is $\mathbf{w}^*$. The precoding vector for each user can be expressed by the following relations

$$\mathbf{w}_{11}^* = \frac{\mathbf{w}^*}{K}, \tag{4.18}$$

$$\mathbf{w}_k = \mathbf{w}_{11}^* e^{j(\varphi_k - \varphi_1)} = \frac{\mathbf{w}^* e^{j(\varphi_k - \varphi_1)}}{K}; \forall\, k = 2, \cdots, K. \tag{4.19}$$

From (4.18), we treat the composite precoding term $\sum_{k=1}^{K} = \mathbf{w}_k e^{j(\varphi_k - \varphi_1)}$ in (2.31) as single vector $\mathbf{w}$, which results in (4.9). If we assume $\mathbf{w}_1^*$ is the optimal solution of the user1‘s precoding vector in (4.9), then without compromising the optimality, the other user‘s precoding vectors are simply the rotated versions of $\mathbf{w}_1^*$ given by

$$\mathbf{w}_i = \mathbf{w}_{11}^* e^{j(\varphi_1 - \varphi_i)}; \ i = 2, \cdots, K. \tag{4.20}$$

It is apparent that the constraint of (4.17) is contained within a hyperslab [162].

4.3.2 Hyperslab Constraints

Given the constraint in (4.17), the precoding vector $\mathbf{w}_1$ is contained within a set of hyperslab $\mathcal{C}$ and also bounded by $\{a, \ b\}$. Therefore, $\mathcal{C}$ is defined as follows

$$\mathcal{C} = \{\mathbf{w}_1 \in \mathbb{R}^{2N_t \times 1}\}|_{a \leq \Lambda^T \Pi \mathbf{w}_1 \leq b}. \tag{4.21}$$

For all $\gamma > 0$ and $\mu > 0$, a proximity barrier function related to (4.21) is given by

$$B(\mathbf{w}_1) = \begin{cases} -\ln(b - \hat{w}_1) - \ln(a + \hat{w}_1), & \text{if } a \leq \hat{w}_1 \leq b \\ +\infty, & \text{otherwise} \end{cases} \tag{4.22}$$

where for convenience, we let $\hat{w}_1 = \Lambda_i^T \Pi \mathbf{w}_1$.

4.3.3 Proximity Operator for the SLP Formulation

To unfold (4.17) into learning framework using IPM, we use its equivalent proximity *'log'* barrier function (4.22) and the proximal operator of $\gamma\mu B(\mathbf{w}_1)$ for every $\mathbf{w}_1$ defined as

$$\Phi(\mathbf{w}_1, \gamma, \mu) = \text{prox}_{\gamma\mu B}(\mathbf{w}_1) = \mathbf{w}_1 + \frac{X(\mathbf{w}_1, \gamma, \mu) - \hat{\Lambda}_i \mathbf{w}_1}{\|\hat{\Lambda}_i\|_2^2} \hat{\Lambda}_i \tag{4.23}$$

where $\hat{\Lambda}_i = \Lambda_i^T \Pi$ and X is a typical solution of the following cubic equation of the form

$$\begin{aligned} x^3 - \left(b + a + \hat{\Lambda}\mathbf{w}_1\right) x^2 + \\ \left(ba + \hat{\Lambda}\mathbf{w}_1 (b + a) - 2\gamma\mu \|\hat{\Lambda}\|_2^2\right) x \\ + \left(-ba\hat{\Lambda}\mathbf{w}_1 + \gamma\mu\,(b + a)\,\|\hat{\Lambda}\|_2^2\right) = 0. \end{aligned} \tag{4.24}$$

It is important to note that the solution to (4.24) is obtained using the analytic solution of the cubic equation. To build the structure of the learning framework, we need to obtain the Jacobian matrix of $\Phi(\mathbf{w}_1, \gamma, \mu)$ with respect to $\mathbf{w}_1$ and the derivatives with respect to γ and μ as follows

$$\mathcal{J}_\Phi \mid_{(\mathbf{w}_1)} = \mathbf{I}_{2N_t} + \frac{1}{\|\hat{\Lambda}_i\|_2^2} \cdot \left(\frac{(b - X(\mathbf{w}_1, \gamma, \mu))\,(a - X(\mathbf{w}_1, \gamma, \mu))}{\Upsilon(\mathbf{w}_1, \gamma, \mu)} - 1 \right) \hat{\Lambda}_i \hat{\Lambda}_i^T, \tag{4.25}$$

$$\Delta_\Phi \mid_{(\mu)} = \frac{-\gamma(b + a - 2X(\mathbf{w}_1, \gamma, \mu))}{\Upsilon(\mathbf{w}_1, \gamma, \mu)} \hat{\Lambda}_i, \tag{4.26}$$

$$\Delta_\Phi \mid_{(\gamma)} = \frac{-\mu\,(b + a - 2X(\mathbf{w}_1, \gamma, \mu))}{\Upsilon(\mathbf{w}_1, \gamma, \mu)} \hat{\Lambda}_i, \tag{4.27}$$

where $\mathbf{I}_{2N_t} \in \mathbb{R}^{2N_t \times 2N_t}$. For hyperslab constraints, $\Upsilon(\cdot)$ is the derivative of (4.24) with respect to x. Finally, using similar abstraction as in subsection 4.3.1, the SLP formulation can be expressed as a succession of sub-problems with respect to the inequality constraint

$$\min_{\mathbf{w_1} \in \mathbb{R}^{2N_t \times 1}} \quad \|\mathbf{w}_1\|_2^2 + \lambda \mathbf{w}_1 + \mu B(\mathbf{w}_1). \tag{4.28}$$

It is important to note that the original problem (4.9) does not have an equality constraint. However, the term $\lambda \mathbf{w}_1$ introduced in (4.28) is to provide additional stability to the network. Using the proximity operator of the barrier, the update rule for every iteration is given by

$$\mathbf{w}_1^{[r+1]} = \text{prox}_{\gamma^{[r]}\mu^{[r]}B}\left(\mathbf{w}_1^{[r]} - \gamma^{[r]}\Delta D(\mathbf{w}_1^{[r]}, \lambda^{[r]})\right) \tag{4.29}$$

where

$$D(\mathbf{w}_1^{[r]}, \lambda^{[r]}) = \|\mathbf{w}_1\|_2^2 + \lambda \mathbf{w}_1, \tag{4.30}$$

and $\Delta\left(D(\mathbf{w}_1^{[r]}, \lambda^{[r]})\right) = \frac{\partial D(\mathbf{w}_1^{[r]}, \lambda^{[r]})}{\partial \mathbf{w}_1^{[r]}}$.

4.3.4 SLP Deep Network (SLP-DNet)

To build the proposed learning-based SLP architecture, we combine an IPM with a proximal forward-backward procedure [163] and transform it into an NN structure represented by the proximity barrier term (see Figure. 4.1). The learning architecture strictly follows the formulation (4.29). We show a striking similarity between our proposal and the feed-forward NN. Intuitively, we form cascade layers of NN from (4.29) as follows

$$\mathbf{w}_1^{[r+1]} = \text{prox}_{\gamma^{[r]}\mu^{[r]}B}\left[\left(\mathbf{I}_{2N_t} - 2\gamma^{[r]}\right)\mathbf{w}_1^{[r]} + \gamma^{[r]}\lambda^{[r]}\mathbf{1}^T\right] \tag{4.31}$$

where $\mathbf{1} \in \mathbb{R}^{1 \times 2N_t}$ is a vector of ones. By letting $\mathbf{W}_r = \mathbf{I}_{2N_t} - 2\gamma^{[r]}$, $\mathbf{b}_r = \gamma^{[r]}\lambda^{[r]}\mathbf{1}^T$ and $\Theta_r = \text{prox}_{\gamma^{[r]}\mu^{[r]}B}$, the r-layer network $\mathcal{L}^{[r-1]} \cdots \mathcal{L}^{[0]}$ will correspond to the fol-

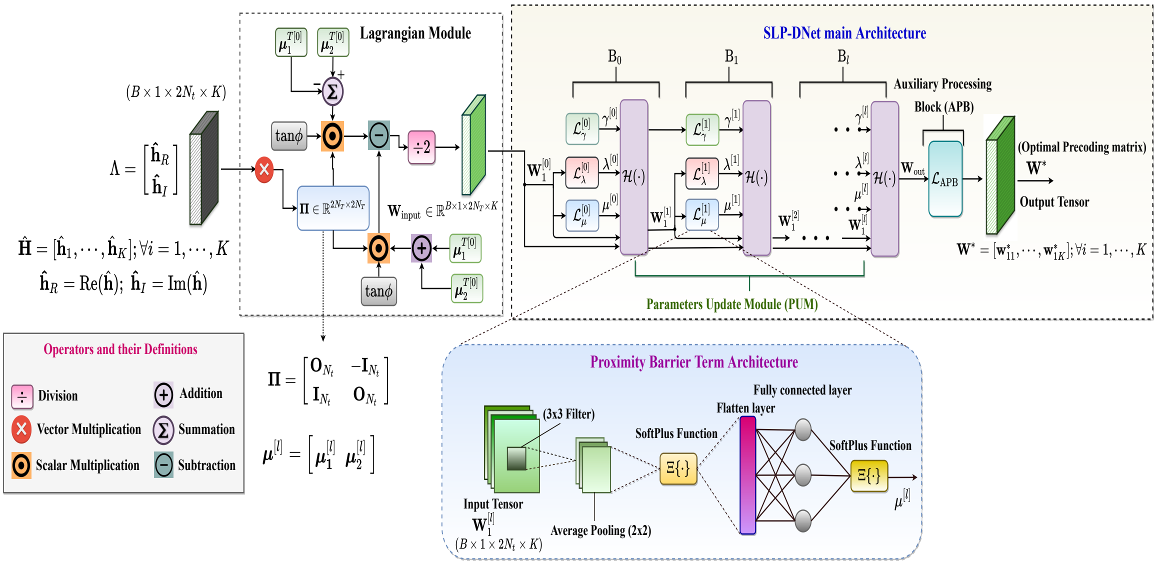

Lagrangian Module
$\mu_1^{T[0]}$
$\mu_2^{T[0]}$
$(B \times 1 \times 2N_t \times K)$
$\Lambda = \begin{bmatrix}\hat{\mathbf{h}}_R \\ \hat{\mathbf{h}}_I\end{bmatrix}$
$\hat{\mathbf{H}} = [\hat{\mathbf{h}}_1, \cdots, \hat{\mathbf{h}}_K]; \forall i = 1, \cdots, K$
$\hat{\mathbf{h}}_R = \mathrm{Re}(\hat{\mathbf{h}});\ \hat{\mathbf{h}}_I = \mathrm{Im}(\hat{\mathbf{h}})$
$\tan\phi$
$\div 2$
$\mathbf{\Pi} \in \mathbb{R}^{2N_T \times 2N_T}$
$\mathbf{W}_{\text{input}} \in \mathbb{R}^{B \times 1 \times 2N_T \times K}$
$\tan\phi$
SLP-DNet main Architecture
B_0
B_1
B_l
Auxiliary Processing Block (APB)
$\mathbf{W}_1^{[0]}$
$\mathcal{L}_\gamma^{[0]}$ $\gamma^{[0]}$
$\mathcal{L}_\lambda^{[0]}$ $\lambda^{[0]}$
$\mathcal{L}_\mu^{[0]}$ $\mu^{[0]}$
$\mathcal{H}(\cdot)$
$\mathbf{W}_1^{[1]}$
$\mathcal{L}_\gamma^{[1]}$ $\gamma^{[1]}$
$\mathcal{L}_\lambda^{[1]}$ $\lambda^{[1]}$
$\mathcal{L}_\mu^{[1]}$ $\mu^{[1]}$
$\mathbf{W}_1^{[2]}$
$\gamma^{[l]}$
$\lambda^{[l]}$
$\mu^{[l]}$
$\mathbf{W}_1^{[l]}$
$\mathbf{W}_{\text{out}}$
$\mathcal{L}_{\text{APB}}$
(Optimal Precoding matrix)
$\mathbf{W}^*$
Output Tensor
$\mathbf{W}^* = [\mathbf{w}_{11}^*, \cdots, \mathbf{w}_{1K}^*]; \forall i = 1, \cdots, K$
Parameters Update Module (PUM)
Operators and their Definitions
÷ Division
⊕ Addition
Vector Multiplication
Σ Summation
⊙ Scalar Multiplication
⊖ Subtraction
$\mathbf{\Pi} = \begin{bmatrix}\mathbf{O}_{N_t} & -\mathbf{I}_{N_t} \\ \mathbf{I}_{N_t} & \mathbf{O}_{N_t}\end{bmatrix}$
$\mu^{[l]} = \begin{bmatrix}\mu_1^{[l]} & \mu_2^{[l]}\end{bmatrix}$
Proximity Barrier Term Architecture
(3x3 Filter)
Fully connected layer
Flatten layer
SoftPlus Function
SoftPlus Function
$\Xi\{\cdot\}$
$\Xi\{\cdot\}$
Input Tensor
$\mathbf{W}_1^{[l]}$
$(B \times 1 \times 2N_t \times K)$
Average Pooling (2x2)
$\mu^{[l]}$

Figure 4.1: Complete SLP-DNet Architecture, showing the parameter update module, the auxiliary processing block

lowing

$$\Theta_0\left(\mathbf{W}_0 + \mathbf{b}_0\right), \cdots, \Theta_r\left(\mathbf{W}_r + \mathbf{b}_r\right) \tag{4.32}$$

where $\mathbf{W}_r$ and $\mathbf{b}_r$ are described as weight and bias parameters respectively. The nonlinear activation functions are defined by Θ_r. In the SLP-DNet design, the Lagrange multiplier associated with the equality constraint is wired across the network to provide additional flexibility. It is important to note that the architectures are the same for both non-robust and robust power minimisation problems described in subsections 4.3 and 4.4 but differ in proximity barrier functions (PBFs). Therefore, to simplify our exposition, we build the structure of the learning framework based on (4.29) and the feed-forward-backward proximal IPM algorithm [159, 163].

As shown in Figure. 4.1, SLP-DNet has two main units; the parameter update module (PUM) and the auxiliary processing block (APB). The PUM has three core components associated with Lagrangian multipliers (equality and inequality constraints) and the training step-size, which are updated according to the following

$$\mathcal{H}(\mathbf{w}_1, \mu, \gamma, \lambda) = \operatorname{prox}_{\gamma^{[r]}\mu^{[r]}B}\left(\mathbf{w}_1^{[r]} - \gamma^{[r]}\Delta D(\mathbf{w}_1^{[r]}, \lambda^{[r]})\right). \tag{4.33}$$

Furthermore, the component that forms the barrier term is constructed with one

Algorithm 1 Proximity Barrier Operator of a Nonrobust SLP-DNet

```
Input: $\mathbf{h}_{Ri}$, $\mathbf{h}_{Ii}$, $\Gamma_i$ and $n_0$ (noise)
Output: $\mathbf{w}_1$, $\gamma$, $\mu$ and $\lambda$
    Initialisation :
 1: Randomly initialise $\mu^{[0]} > 0$, $\lambda^{[0]} > 0$, $\gamma^{[0]} > 0 \; \forall \, i = 1, \cdots, K$ and $\mathbf{w}_0 \in \mathbb{R}^{2N_t \times 1}$
    using (4.36)
 2: Find the solution to (4.24) using Cardano formula.
 3: For every solution in step 2, compute its corresponding Barrier function using
    (4.22).
 4: Compute the Proximity Operator of the Barrier at $\mathbf{w}_0$ using (4.13), where
    $\Phi(\mathbf{w}_1, \gamma, \mu) = \text{prox}_{\gamma\mu B}(\mathbf{w}_1)$.
 5: Compute the derivatives of the Proximity Operator w.r.t $\mathbf{w}_1$, $\mu$ and $\gamma$ using
    (4.25), (4.26) and (4.27).
 6: for $r = 0$ to $L$ do
 7:     Update the training variables as follows:
            (a) $\mu^{[r+1]} = \mu^{[r]} - \eta \frac{\partial \Phi(\mathbf{w}_1^{[r]}, \gamma^{[r]}, \mu^{[r]})}{\partial \mu^{[r]}}$
            (b) $\gamma^{[r+1]} = \gamma^{[r]} - \eta \frac{\partial \Phi(\mathbf{w}_1^{[r]}, \gamma^{[r]}, \mu^{[r]})}{\partial \gamma^{[r]}}$
            (c) $\lambda^{[r+1]} = \lambda^{[r]} - \eta \frac{\partial D(\mathbf{w}_1^{[r]}, \lambda^{[r]})}{\partial \lambda^{[r]}}$ using (4.30)
        where $\eta$ is the learning rate.
        Feed-forward-Backward Proximal IPM
 8:     $\mathbf{w}_1^{[r+1]} = \text{prox}_{\gamma^{[r]}\mu^{[r]}B}\left(\mathbf{w}_1^{[r]} - \gamma^{[r]} \Delta D(\mathbf{w}_1^{[r]}, \lambda^{[r]})\right)$
 9: end for
10: return $\mathbf{w}_1^*$ (Optimal precoding tensor).
11: To obtain the original optimal complex precoding vector $\mathbf{w}^*$, we use the relation
    $\mathbf{w}_1^* = [\mathbf{w}_R^* \; - \mathbf{w}_I^*]$ to separate it into real and imaginary parts.
```

convolutional layer, an average pooling layer, a fully connected layer, and a Softplus layer to curb the output to a positive real value to satisfy the inequality constraint. The APB unit is connected to the last r-th block of the PUM in the form of a deep CNN to convert the output of the last parameter update block into a target transmit precoding vector. The APB architecture is made up of 3 convolution layers and 2 activation layers. In addition, a Batch Normalisation layer is added between each convolutional layer and the activation layer to stabilise the mismatch in the distribution of the inputs caused by the internal covariate shift [164]. For every r block (r-th layer), there are three core components; $\mathcal{L}_{\mu}^{[r]}$, $\mathcal{L}_{\gamma}^{[r]}$ and $\mathcal{L}_{\lambda}^{[r]}$ associated with the learnable parameters (μ, γ and λ), respectively as shown in Figure 4.1. These components form a learning block for computing the barrier parameter (μ) associated with the inequality constraint, the step-size (γ) and the equality constraint (λ), if exists.

To ensure that the constraints remain positive, a Softplus-sign function [165], $\text{Softplus}(z) = \ln(1+\exp(z))$ is used. The Softplus-sign function is a smooth approximation of the rectified linear unit (**ReLu**) activation function; and unlike the **ReLu** its gradient is never exactly equal to zero [165], which imposes an update on γ, μ and λ during the backward propagation. The nonrobust SLP-DNet formulation and its training steps are summarised in Algorithm 1. The training variables are updated iteratively in each unfolding layer simultaneously using gradient descent (GDS). For every training step, a corresponding value of the precoding vector is updated using (4.29). The output precoding vector from the last PUM is then fed into the APB to obtain the final optimal precoding vector. The same algorithm is also adopted for a robust SLP-DNet but with a different PBF based on a robust power minimisation problem.

Finally, the output from the APB is the precoding vector in the real domain. The relation: $\mathbf{w}_1 = [\mathbf{w}_R \; -\mathbf{w}_I]^T$ is used to convert it to its equivalent complex domain for every SINR value of the i-th user. Finally, the NN structures of the PBF term and the APB are summarised in Tables 4.1 and 4.2.

Table 4.1: Proximity Barrier Function NN Layout

Layer	Parameter, kernel size $= 3\times 3$
Input Layer	Input size $(\text{B},\ 1,\ 2N_t,\ K)$
Layer 1: Convolutional	Size $(\text{B}, 1, K, 20)$; zero padding
Layer 2: Average Pooling	Size $((1,\ 1),\ \text{stride} = (1,\ 1))$
Layer 3: Activation	Soft-Plus
Layer 4: Flat	Size $(\text{B}\times 20\times K^2)$
Layer : Fully-connected	Size$(\text{B}\times 20\times K^2,\ 1)$
Layer 5: Activation	Soft-Plus function

4.3.5 Duality and Loss Function of the SLP Formulation

In order to ease the formulation of the dual-problem of the original problem (4.9), the left-hand-side of the inequality constraint is split into its equivalent positive and

Table 4.2: Auxiliary Processing Block (APB) NN Structure

Layer	Parameter, kernel size $= 3 \times 3$
Input Layer	Input size $(\mathrm{B},\ 1,\ 2N_t,\ K)$
Layer 1: Convolutional	Size $(\mathrm{B},\ 1,\ K,\ 64)$, dilation $= 1$ and unit padding
Layer 2: Batch Normalisation	eps $= 10^{-6}$, momentum $= 0.1$
Layer 3: Activation	PReLu
Layer 4: Convolutional	Size $(\mathrm{B},\ 1,\ 64,\ 2N_tK)$, dilation $= 1$ and unit padding
Layer 5: Batch Normalisation	eps $= 10^{-6}$, momentum $= 0.1$
Layer 6: Activation	PReLu
Layer 7: Convolutional	Size $(\mathrm{B},\ 1,\ 2N_tK,\ 1)$, dilation $= 1$ and unit padding

negative parts as follows

$$\begin{aligned} \min_{\{\mathbf{w}_1\}} \quad & \|\mathbf{w}_1\|_2^2 \\ \text{s.t.} \quad & \Lambda_i^T \Pi \mathbf{w}_1 \leq \left(\Lambda_i^T \Pi \mathbf{w}_1 - \sqrt{\Gamma_i n_0} \right) \tan\phi,\ \forall i \\ & -\Lambda_i^T \Pi \mathbf{w}_1 \leq \left(\Lambda_i^T \Pi \mathbf{w}_1 - \sqrt{\Gamma_i n_0} \right) \tan\phi,\ \forall i. \end{aligned} \tag{4.34}$$

The Lagrangian of (4.34) is defined as

$$\begin{aligned} \mathcal{L}_{\mathrm{rl}}(\mathbf{w}_1,\ \mu_1,\ \mu_2) = \|\mathbf{w}_1\|_2^2 + \mu_1 \left(\Lambda_i^T \Pi \mathbf{w}_1 - \Lambda_i^T \mathbf{w}_1 \tan\phi + \sqrt{\Gamma_i n_0} \right) \\ - \mu_2 \left(\Lambda_i^T \Pi \mathbf{w}_1 + \Lambda_i^T \mathbf{w}_1 \tan\phi - \sqrt{\Gamma_i n_0} \right), \end{aligned} \tag{4.35}$$

where μ_1 and μ_2 are the Lagrangian multipliers associated with the constraints and are related to the proximity barrier. The subscript *'rl'* stands for relaxed phase rotation. It can be easily proven that the lower bound (**LB**) of (4.35) is $\mathcal{L}_{\mathrm{rl}}(\mathbf{w}_1, \mu_1, \mu_2) \geq \mu_1 \Lambda_i (\Pi - \tan\phi) - \mu_2 \Lambda_i (\Pi + \tan\phi)$. From (4.35), the optimal precoder is obtained by differentiating $\mathcal{L}_{\mathrm{rl}}(\cdot)$ w.r.t $\mathbf{w}_1$ and equating to zero. By doing so, the optimal precoder can be found as

$$\mathbf{w}_1 = \frac{\left({\mu_1}^T + {\mu_2}^T \right) \cdot \Lambda_i \tan\phi - \left({\mu_1}^T - {\mu_2}^T \right) \cdot \Pi^T \Lambda_i \tan\phi}{2}. \tag{4.36}$$

In the sequel, we show that (4.36) is used to generate the training input (precoding vector) by randomly initialising the Lagrangian multipliers (μ_1 and μ_2) and then train the network to learn their values that minimise the loss function (Lagrangian function). The loss function is modified by adding l_2-norm regularisation over the weights to calibrate the learning coefficients in order to adjust the learning process. It should be noted that the regularisation here is not aimed at addressing the problem of overfitting as in the case of supervised learning. However, regularisation in an unsupervised learning is used to normalise and moderate weights attached to a neuron to help stabilise the learning algorithm [166]. The loss function (4.35) over N training samples is thus expressed as

$$\begin{aligned}\mathcal{L}_{\mathrm{rl}}(\mathbf{w}_1,\, \mu_1,\, \mu_2) = {} & \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{w}_1\|_2^2 + \frac{1}{N}\sum_{i=1}^{N}\left(\mu_1\left(\Lambda_i^T\Pi\mathbf{w}_1 - \Lambda_i^T\mathbf{w}_1\tan\phi + \sqrt{\Gamma_i n_0}\right)\right) \\ & - \frac{1}{N}\sum_{i=1}^{N}\left(\mu_2\left(\Lambda_i^T\Pi\mathbf{w}_1 + \Lambda_i^T\mathbf{w}_1\tan\phi - \sqrt{\Gamma_i n_0}\right)\right) + \frac{\vartheta}{NL}\sum_{i=1}^{N}\sum_{i=1}^{L}\|\theta_i\|_2^2, \end{aligned} \tag{4.37}$$

where θ_i are the trainable parameters of the i-th layers associated with the weights and biases, and $\vartheta > 0$ is the penalty parameter that controls the bias and variance of the trainable coefficients, N, L is the number of training samples (batch size or the number of channel realisation) and the number of layers, respectively.

4.3.6 Learning-Based SLP for Strict Angle Rotation

In the previous subsection, we have presented SLP-DNet based on relaxed angle formulation. In this subsection, we provide a formulation for strict phase angle rotation where all the interfering signals align exactly to the phase of the signal of interest (i.e. $\phi = 0$ in Figure 2.4), the optimisation problem is [49]

$$\begin{aligned} \min_{\{\mathbf{w}_1\}} \quad & \|\mathbf{w}_1\|^2 \\ \text{s.t.} \quad & \Lambda_i^T\Pi\mathbf{w}_1 = 0\ ,\ \forall i \\ & \Lambda_i^T\mathbf{w}_1 \geq \sqrt{\Gamma_i n_0}\ ,\ \forall i. \end{aligned} \tag{4.38}$$

We observe that the inequality constraint in (4.38) is affine. Based on this, the proximal barrier function for the strict phase rotation is

$$B_{\mathrm{st}}(\mathbf{w}_1) = \begin{cases} -\ln\left(\Lambda_i^T \mathbf{w}_1 - \sqrt{\Gamma_i n_0}\right), & \text{if } \Lambda_i^T \mathbf{w}_1 \geq \sqrt{\Gamma_i n_0} \\ +\infty, & \text{otherwise.} \end{cases} \tag{4.39}$$

The subscript *'st'* represents strict phase rotation. Therefore, for every precoding vector $\mathbf{w}_1 \in \mathbb{R}^{2N_t \times 1}$, the proximity operator of $\mu\gamma B_{\mathrm{st}}$ at $\mathbf{w}_1$ is given by

$$\Phi_{\mathrm{st}}(\mathbf{w}_1, \mu, \gamma) = \mathbf{w}_1 + \frac{\Lambda_i{}^T \mathbf{w}_1 - \sqrt{\Gamma_i n_0} - \sqrt{(\Lambda_i^T \mathbf{w}_1 - \sqrt{\Gamma_i n_0})^2 + 4\gamma\mu \|\Lambda_i^T\|_2^2}}{2\|\Lambda_i\|_2^2} \Lambda_i. \tag{4.40}$$

Similar to the steps in subsection 4.3.3, the learning-based framework for SLP strict phase rotation is designed by finding the Jacobian matrix of $\Phi(\mathbf{w}_1, \mu, \gamma)$ with respect to $\mathbf{w}_1$, and the derivatives of $\Phi(\mathbf{w}_1, \mu, \gamma)$ with respect to γ and μ can be easily obtained from (4.40). The loss function over N training batches is given by

$$\begin{aligned} \mathcal{L}_{\mathrm{st}}(\mathbf{w}_1, \lambda,\ \mu) = \frac{1}{N} \sum_{i=1}^{N} \left(\|\mathbf{w}_1\|_2^2 + \lambda \Lambda_i^T \Pi \mathbf{w}_1\right) + \\ \frac{1}{N} \sum_{i=1}^{N} \left(\mu \left(\sqrt{\Gamma_i n_0} - \Lambda_i^T \mathbf{w}_1\right)\right) + \frac{\vartheta}{NL} \sum_{i=1}^{N} \sum_{i=1}^{L} \|\theta_i\|_2^2, \end{aligned} \tag{4.41}$$

where μ and λ are the Lagrangian multipliers for inequality and equality constraints, respectively. Finally, minimising (4.41) with respect to $\mathbf{w}_1$ (differentiating $\mathcal{L}_{\mathrm{st}}(\cdot)$ w.r.t $\mathbf{w}_1$), gives the initial training optimal precoder as

$$\mathbf{w}_1 = \frac{\mu^T \cdot \Lambda_i - \lambda^T \cdot \Pi \Lambda_i}{2}. \tag{4.42}$$

4.4 Learning Robust Power Minimisation SLP with Channel Uncertainty

So far, we have derived the unsupervised learning scheme in which the uncertainty in estimating the channel coefficients is not considered.

4.4.1 Robust Optimisation-Based SLP Formulation

The multi-cast CI formulation of the power minimisation problem for the worst-case CSI error based on (4.9) is given by [77]

$$\begin{aligned} \min_{\{\mathbf{w}\}} \quad & \|\mathbf{w}_2\|^2 \\ \text{s.t.} \quad & \left|\mathrm{Im}\left(\hat{\mathbf{h}}_i^T\mathbf{w}\right)\right| - \left(\mathrm{Re}\left(\hat{\mathbf{h}}_i^T\mathbf{w}\right) - \sqrt{\Gamma_i n_0}\right)\tan\phi \le 0, \\ & \forall \|\hat{\mathbf{e}}\|_i{}^2 \le \varsigma_i^2,\ \forall i. \end{aligned} \tag{4.43}$$

The intractability of the constraint in (4.43) can be effectively handled using convex optimisation methods. Therefore, to guarantee that the robust constraint in (4.43) is satisfied, it is modified as follows

$$\max_{\|\bar{e}_i\|^2 \le \varsigma_i^2} \left(\left|\mathrm{Im}\left(\hat{\mathbf{h}}^T\mathbf{w}\right)\right| - \left(\mathrm{Re}\left(\hat{\mathbf{h}}^T\mathbf{w}\right) - \sqrt{\Gamma n_0}\right)\tan\phi\right) \le 0. \tag{4.44}$$

It is worth noting that the subscripts in (4.44) are ignored in order to simplify the problem formulation. By defining the equivalent real-valued channel and channel error vectors, the real and imaginary parts in the constraint can be decomposed into two separate constraints as explained in Section 4.3 (see (4.6a) and (4.6b)). Thus the robust formulation of the constraint is equivalent to two separate real-valued constraints as follows

$$\Lambda^T\mathbf{w}_1 - \Lambda^T\mathbf{w}_2\tan\phi + \varsigma\|\mathbf{w}_1 - \mathbf{w}_2\tan\phi\|_2 + \sqrt{\Gamma n_0}\tan\phi \le 0, \tag{4.45}$$

$$-\Lambda^T\mathbf{w}_1 - \Lambda^T\mathbf{w}_2\tan\phi + \varsigma\|\mathbf{w}_1 + \mathbf{w}_2\tan\phi\|_2 + \sqrt{\Gamma n_0}\tan\phi \le 0, \tag{4.46}$$

where $\Lambda = \begin{bmatrix}\bar{\mathbf{h}}_R & \bar{\mathbf{h}}_I\end{bmatrix}^T$, $\mathbf{e} \triangleq \begin{bmatrix}\bar{\mathbf{e}}_R & \bar{\mathbf{e}}_I\end{bmatrix}^T$ and $\hat{\mathbf{h}} = \bar{\mathbf{h}}_R + j\bar{\mathbf{h}}_I + \bar{\mathbf{e}}_R + j\bar{\mathbf{e}}_I$. Finally, the robust CI formulation for power minimization problem becomes

$$\begin{aligned} \min_{\{\mathbf{w_1},\mathbf{w_2}\}} \quad & \|\mathbf{w}_1\|_2^2 \\ \text{s.t.} \quad & \text{Constraints (4.45) and (4.46)},\ \forall i \\ & \text{where } \mathbf{w}_1 = \Pi\mathbf{w}_2. \end{aligned} \tag{4.47}$$

4.4.2 Unsupervised Learning-Based Robust SLP Formulation

In this subsection, we extend our proposed unsupervised learning formulation to a worst-case CSI-error to design a robust precoding scheme for the power minimisation problem. As an extension of the previous formulations in subsection 4.3.3, the focus here is to derive a PBF for the robust learning-based precoding scheme. Substituting for $\mathbf{w}_1$ in (4.47), we have

$$\left(\Lambda^T \Pi - \Lambda^T \tan\phi\right) \mathbf{w}_2 + \varsigma \|(\Pi - \tan\phi)\, \mathbf{w}_2\|_2 + \sqrt{\Gamma n_0} \tan\phi \leq 0, \tag{4.48}$$

$$-\left(\Lambda^T \Pi + \Lambda^T \tan\phi\right) \mathbf{w}_2 + \varsigma \|(\Pi + \tan\phi)\, \mathbf{w}_2\|_2 + \sqrt{\Gamma n_0} \tan\phi \leq 0. \tag{4.49}$$

Apparently, the constraints (4.48) and (4.49) are bounded by the l_2-norm. Therefore, problem (4.47) becomes

$$\begin{aligned} \min_{\{\mathbf{w}_2\}} \quad & \|\mathbf{w}_2\|_2^2 \\ \text{s.t.} \quad & \text{Constraints (4.48) and (4.49)}, \ \forall i. \end{aligned} \tag{4.50}$$

The resulting barrier function of the corresponding constraints of (4.50) is the sum of the individual barrier functions associated with the two inequality constraints. We begin by introducing the feasible set of solutions bounded by the Euclidean ball.

4.4.3 Bounded Euclidean ball Constraint

Suppose a problem whose set of feasible solutions is bounded by the Euclidean ball [167]

$$\mathcal{C} = \{\mathbf{z} \in \mathbb{R}^n \big| \|\mathbf{z} - \mathbf{x}\|_2 \leq \beta\}, \tag{4.51}$$

where $\beta > 0$ and $\mathbf{x} \in \mathbb{R}^n$. Let $\gamma > 0$ and $\mu > 0$ be the step-size and the Lagrange multiplier associated with the inequality constraint, respectively. Then the barrier function is expressed as [167]

$$B(\mathbf{z}) = \begin{cases} -\ln(\beta - \|\mathbf{z} - \mathbf{x}\|_2), & \text{if } \|\mathbf{z} - \mathbf{x}\|_2 < \beta, \\ +\infty, & \text{otherwise} \end{cases} \tag{4.52}$$

For simplicity, we let $\mathbf{Q}_1 = (\Pi - \mathbf{I}_{2N_t}\tan\phi)$ and $\mathbf{Q}_2 = (\Pi + \mathbf{I}_{2N_t}\tan\phi)$. Based on (4.52), the barrier function corresponding to the constraint (4.48) is

$$B_1(\mathbf{w}_2) = \begin{cases} -\ln\left(-\sqrt{\Gamma n_0}\tan\phi - (\Lambda^T\mathbf{Q}_1\mathbf{w}_2 + \varsigma||\mathbf{Q}_1\mathbf{w}_2||_2)\right), & \text{if } \Lambda^T\mathbf{Q}_1\mathbf{w}_2 + \varsigma||\mathbf{Q}_1\mathbf{w}_2||_2 < -\sqrt{\Gamma n_0}\tan\phi \\ +\infty & \text{otherwise} \end{cases} \tag{4.53}$$

In the case of constraint (4.49), similar expression is also written for $B_2(\mathbf{w}_2)$ using $\mathbf{Q}_2$ as in (4.54)

$$B_2(\mathbf{w}_2) = \begin{cases} -\ln\left(-\sqrt{\Gamma n_0}\tan\phi - (\Lambda^T\mathbf{Q}_2\mathbf{w}_2 + \varsigma||\mathbf{Q}_2\mathbf{w}_2||_2)\right), & \text{if } \Lambda^T\mathbf{Q}_2\mathbf{w}_2 + \varsigma||\mathbf{Q}_2\mathbf{w}_2||_2 < -\sqrt{\Gamma n_0}\tan\phi \\ +\infty, & \text{otherwise,} \end{cases} \tag{4.54}$$

. The resulting barrier function is thus expressed in (4.55)

$$B_{\text{robust}}(\mathbf{w}_2) = B_1(\mathbf{w}_2) + B_2(\mathbf{w}_2). \tag{4.55}$$

Without loss of generality, the constraints (4.48) and (4.49) can be further written as

$$\Lambda^T\mathbf{Q}_1\mathbf{w}_2 + \varsigma||\mathbf{Q}_1\mathbf{w}_2||_2 + \sqrt{\Gamma n_0}\tan\phi \leq 0, \tag{4.56}$$

$$\Lambda^T\mathbf{Q}_2\mathbf{w}_2 + \varsigma||\mathbf{Q}_2\mathbf{w}_2||_2 + \sqrt{\Gamma n_0}\tan\phi \leq 0. \tag{4.57}$$

It can be seen that the upper bound of the two constraints (4.56) and (4.57) is zero, Therefore, the effective proximity operator of (4.55) is obtained the by squaring (4.56) and (4.57) and adding the results. Following similar steps presented in subsection 4.3.3, we obtain the proximity operator of the barrier for the robust SLP-DNet (see Appendix A for details).

4.4.4 Loss Function of the Robust Power Minimisation Problem

The training loss function is the Lagrangian of (4.50), and can be written as

$$\begin{aligned} \min_{\{\mathbf{w}_2\}} \quad & ||\mathbf{w}_2||_2^2 \\ \text{s.t.} \quad & \Lambda^T\mathbf{Q}_1\mathbf{w}_2 + \varsigma||\mathbf{Q}_1\mathbf{w}_2||_2 + \sqrt{\Gamma n_0}\tan\phi \leq 0\ \forall i \\ & \Lambda^T\mathbf{Q}_2\mathbf{w}_2 + \varsigma||\mathbf{Q}_2\mathbf{w}_2||_2 + \sqrt{\Gamma n_0}\tan\phi \leq 0\ \forall i. \end{aligned} \tag{4.58}$$

Therefore, the loss function of (4.58) is the regularised Lagrangian parameterised by θ_i over the entire layers

$$\begin{aligned}\mathcal{L}_{\text{robust}}(\mathbf{w}_2,\ \mu_1,\ \mu_2) = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{w}_2\|_2^2 \\ + \frac{\mu_1}{N}\sum_{i=1}^{N}\left(\varsigma^2\|\mathbf{Q}_1\mathbf{w}_2\|_2^2 - \left(\sqrt{\Gamma n_0}\tan\phi - \Lambda^T\mathbf{Q}_1\mathbf{w}_2\right)^2\right) \\ + \frac{\mu_2}{N}\sum_{i=1}^{N}\left(\varsigma^2\|\mathbf{Q}_2\mathbf{w}_2\|_2^2 - \left(\sqrt{\Gamma n_0}\tan\phi - \Lambda^T\mathbf{Q}_2\mathbf{w}_2\right)^2\right) + \frac{\vartheta}{NL}\sum_{i=1}^{N}\sum_{i=1}^{L}\|\theta_i\|_2^2.\end{aligned} \tag{4.59}$$

The minimum of (4.59) with respect to $\mathbf{w}_2$ is obtained by equating its derivative to zero

$$\left(1 + \left(\mu_1\|\mathbf{Q}_1\|_2^2 + \mu_2\|\mathbf{Q}_2\|_2^2\right)\left(\varsigma^2 - \Lambda^T\Lambda\right)\right)\mathbf{w}_2 = -\left(\mu_1\mathbf{Q}_1 + \mu_2\mathbf{Q}_2\right)\Lambda\sqrt{\Gamma n_0}\tan\phi. \tag{4.60}$$

For convenience, we redefine the real matrices and vectors as $\begin{bmatrix}\|\mathbf{Q}_1\|_2^2 & \|\mathbf{Q}_2\|_2^2\end{bmatrix} = \bar{\mathbf{q}}_{\text{norm}}$; $\begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix} = \bar{\mathbf{Q}}$ and $\begin{bmatrix}\mu_1 & \mu_2\end{bmatrix} = \bar{\mu}$. With these new notations, (4.60) is simplified to

$$\left(\mathbf{I}_{2N_t} + \bar{\mathbf{q}}_{\text{norm}}\bar{\mu}^T\left(\varsigma^2 - \Lambda^T\Lambda\right)\right)\mathbf{w}_2 = -\Lambda\bar{\mathbf{Q}}\bar{\mu}^T\sqrt{\Gamma n_0}\tan\phi. \tag{4.61}$$

From (4.61), the initial training optimal transmit precoder is thus

$$\mathbf{w}_2 = -\Lambda\bar{\mathbf{Q}}\bar{\mu}^T\mathbf{A}^{-1}\sqrt{\Gamma n_0}\tan\phi, \tag{4.62}$$

where $\mathbf{A} = \left(\mathbf{I}_{2N_t} + \bar{\mathbf{q}}_{\text{norm}}\bar{\mu}^T\left(\varsigma^2 - \Lambda^T\Lambda\right)\right)$. Note that the Lagrange multipliers μ_1 and μ_2 are associated with the barrier term and are randomly initialised from a uniform distribution.

4.5 Data Generation

The channel coefficients are used to form a dataset and are generated randomly from a normal distribution with zero mean and unit variance. The data input tensor is obtained using (4.2). We summarise the entire dataset preprocessing procedure

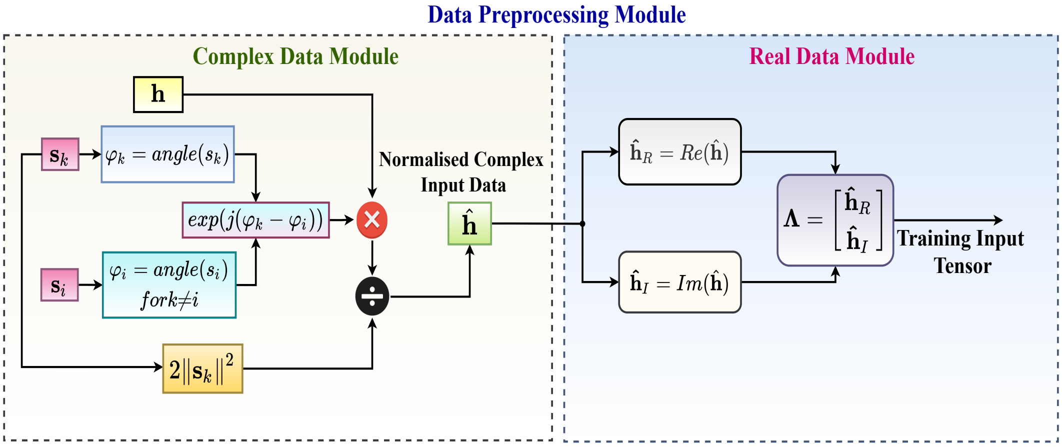

Data Preprocessing Module
Complex Data Module
Real Data Module
$\mathbf{h}$
$\mathbf{s}_k$
$\varphi_k = angle(s_k)$
$exp(j(\varphi_k - \varphi_i))$
$\mathbf{s}_i$
$\varphi_i = angle(s_i)$
$fork{\neq}i$
$2\|\mathbf{s}_k\|^2$
Normalised Complex Input Data
$\hat{\mathbf{h}}$
$\hat{\mathbf{h}}_R = Re(\hat{\mathbf{h}})$
$\hat{\mathbf{h}}_I = Im(\hat{\mathbf{h}})$
$\mathbf{\Lambda} = \begin{bmatrix} \hat{\mathbf{h}}_R \\ \hat{\mathbf{h}}_I \end{bmatrix}$
Training Input Tensor

Figure 4.2: Dataset Generating Block

in Figure. 4.2. It can be observed that the input dataset is normalised by the transmit data symbol so that data entries are within the nominal range, and this could potentially aid the training.

4.5.1 SLP-DNet Training and Inference

The training of DNN generally involves three steps: forward propagation, backward propagation, and parameter update [130]. Except where necessary, the training SINR is drawn from a random uniform distribution to enable learning over a wide range of SINR values. The PUM contains r blocks, which form a learning layer. Therefore, each block contains three core components and is trained block-wise for l number of iterations.

Similarly, the APB is trained for k iterations. It is important to note that the number of training iterations of the parameter update module may not necessarily be equal to that of the APB. We train the PUM for 15 iterations and the APB for 10 iterations. To improve the training efficiency, we modify the learning rate by a factor $\alpha \in \mathbb{R}^+$ for every training step. All the training is done with a stochastic gradient descent algorithm using Adam optimiser [130]. Since the learning is done in an unsupervised fashion, the loss function is the Lagrangian function's statistical mean over the entire training batch samples. During the inference, a feed-forward pass is performed over the entire architecture using the learned Lagrangian multipliers to calculate the precoding vector using (4.36) and (4.62) for both SLP and robust

SLP formulations, respectively. Finally, at inference, the trained model is run with different SINR values to obtain the required optimal precoding matrix.

4.6 Computational Complexity Evaluation

In this subsection, we analyse and compare the computational costs of the conventional BLP, optimisation-based SLP, and the proposed SLP-DNet schemes. The complexities are evaluated in terms of the number of real arithmetic operations involved. For ease of analysis, we convert the SOCP (4.9) into a standard linear programming (LP)

$$\begin{aligned} \min_{\{\mathbf{z}\}} \quad & \mathbf{c}^T \mathbf{z} \\ \text{s.t.} \quad & \mathbf{c}_k^T \mathbf{z} \leq -\tan\phi \sqrt{\Gamma_i n_0} \text{ , } \forall i \end{aligned} \tag{4.63}$$

where $\mathbf{c} = \begin{bmatrix} 0 & \mathbf{w}_1^T \end{bmatrix}^T \in \mathbb{R}^{(2N_t+1)\times 1}$, $\mathbf{z} = \begin{bmatrix} 1 & \mathbf{w}_1 \end{bmatrix}^T \in \mathbb{R}^{(2N_t+1)\times 1}$, $\mathbf{c}_k = \left[\left| \Lambda_i^T \Pi \mathbf{w}_1 \right| \; \Lambda_i^T \tan\phi \right]^T \in \mathbb{R}^{(2N_t+1)\times 1}$ and $\mathbf{W} = [\mathbf{w}_{11}, \cdots, \mathbf{w}_{1K}]$; $\forall i = 1, \cdots, K$. The complexity per iteration for solving convex optimisation via IPM is dominated by the formation ($\mathtt{C}_{\text{form}}$) and factorisation ($\mathtt{C}_{\text{fact}}$) of the matrix coefficients of m linear equations in m unknowns [168]. For generic IPMs, the complexity is expressed as [168]

$$\mathtt{C}_{\text{total}} = \mathtt{C}_{\text{iter}} \cdot \left(\mathtt{C}_{\text{form}} + \mathtt{C}_{\text{fact}} \right) \tag{4.64}$$

where $\mathtt{C}_{\text{iter}}$ is the iteration complexity required to attain an optimal solution. For a given optimal target accuracy, $\varepsilon > 0$, $\mathtt{C}_{\text{iter}}$ is given by

$$\mathtt{C}_{\text{iter}} = \sqrt{\sum_{j=1}^{N_{\text{lc}}} d_j + 2N_{\text{sc}}} \times \ln\left(\frac{1}{\varepsilon}\right) \tag{4.65}$$

where d is the dimension of the constraints, N_{lc} and N_{sc} are the numbers of linear inequality matrix and second order cone (SOC) constraints, respectively. The costs of formation and factorisation of matrix are respectively given by [168]

$$\mathtt{C}_{\text{form}} = \underbrace{m \sum_{j=1}^{N_{\text{lc}}} d_j^3 + m^2 \sum_{j=1}^{N_{\text{lc}}} d_j^2}_{\text{due to } N_{\text{lc}}} + \underbrace{m \sum_{j=1}^{N_{\text{sc}}} d_{j=1}^2}_{\text{due to } N_{\text{sc}}}; \; \mathtt{C}_{\text{fact}} = m^3. \tag{4.66}$$

Specifically, we observe that problem (4.63) has K constraints with dimension $2N_t + 1$. Therefore, using (4.65) and (4.66), the total computational complexity is thus

$$\mathsf{C}_{\text{total}} = \sqrt{2N_t + 1}\left[m(2N_t+1) + m(2N_t+1)^2 + m^3\right] \ln\left(\tfrac{1}{\varepsilon}\right). \tag{4.67}$$

The complexity of BLP can be derived in a similar way and is shown directly in Table 4.3. Conversely, the complexity of the proposed SLP-DNet schemes is the sum of PUM and the APB complexities. Moreover, the complexity of the PUM is dominated by the costs of computing the *'log barrier'* and the feed-forward pass of the shallow CNN (see Table 4.1) that makes up the barrier term associated with the inequality constraint. Similarly, the complexity of the APB is also obtained by computing the arithmetic operations involved during the forward pass of the deep CNN (see Table 4.2). To derive the analytical complexity of SLP-DNet, we assume a sliding window is used to perform the dominant computation of the convolution operation in the CNN and ignore the nonlinear computational overhead due to activations. Therefore, the total computational complexity is expressed as

$$\begin{aligned} C_{SLP-DNet} = {} & C_{\text{log-br}} + \\ & 2\sum_{l=1}^{L_{\text{conv}}} n_{\text{h}}^{[l-1]} n_{\text{w}}^{[l-1]} \left[C_{\text{in}}^{[l-1]} f^{[l]2} + 1\right] C_{\text{out}}^{[l]} + \sum_{j=1}^{L_{\text{fc}}} \left(2M_{\text{in}}^{[j-1]} + 1\right) M_{\text{out}}^{[i]} \end{aligned} \tag{4.68}$$

where n_{h}, n_{w}, f, C_{in} and C_{out} are the height, width of the input tensor, kernel size, number of input and output channels, respectively. Similarly, L_{conv}, L_{fc}, M_{in} and M_{out} are the number of convolution and fully connected (FC) layers, number of input and output neurons in the FC layer, respectively. $C_{\text{log-br}}$ denotes the complexity of the *'log-barrier'* function. Table 4.3 shows the summary of the computational complexities of our proposals and the benchmark precoding schemes. As an illustration, we consider the case of a symmetrical system $(N_t = K = n)$, and show that the proposed approach has substantially reduced computational complexity of $\mathscr{O}(n^3)$, while the optimisation-based SLP approach of $\mathscr{O}(n^{6.5})$ and the conventional BLP is $\mathscr{O}(n^{7.5})$.

Table 4.3: Complexity analysis of proposed SLP-DNet and benchmark SLP schemes.

Problem	Arithmetic Operations (term; $m = \mathcal{O}(2N_tK)$)	Complexity Order ($n = N_t = K$)
Conventional BLP	$\sqrt{(4N_t+K+2)}\left[m(2N_t+1)+m(2N_t+1)^2+m(K+1)^2+m^3\right]\ln\left(\frac{1}{\varepsilon}\right)$	$\mathcal{O}(n^{6.5})$
SLP Optimisation-based	$\sqrt{2N_t+1}\left[m(2N_t+1)+m(2N_t+1)^2+m^3\right]\ln\left(\frac{1}{\varepsilon}\right)$	$\mathcal{O}(n^{6.5})$
SLP-DNet	$4K^2N_t+42K^2+48KN_t+512K+2$	$\mathcal{O}(n^3)$
SLP-DNet Strict	$4K^2N_t+39K^2+46KN_t+512K+2$	$\mathcal{O}(n^3)$
Robust Conventional BLP	$\sqrt{2K(2N_t+1)}\left[mK(2N_t+1)^3+m^2K(2N_t+1)^2+m^3\right]\ln\left(\frac{1}{\varepsilon}\right)$	$\mathcal{O}(n^{7.5})$
Robust SLP Optimisation-based	$\sqrt{2(2N_t+1)}\left[2mK(2N_t+1)^2+m^3\right]\ln\left(\frac{1}{\varepsilon}\right)$	$\mathcal{O}(n^{6.5})$
Robust SLP-DNet	$16KN_t^2+42K^2+48KN_t+512K$	$\mathcal{O}(n^3)$

Table 4.4: Simulation parameters

Parameters	Values
Training Samples	50000
Batch Size (B)	200
Test Samples	2000
Training SINR range	0.0dB - 45.0dB
Test SINR range (i-th user SINR)	0.0dB - 35.0dB
Optimiser	SGD with Adam
Initial Learning Rate η	0.001
Learning Rate decay factor α	0.65
Weight Initialiser	Xavier Initialiser
Number of blocks in the parameter update unit	$B_r = 3$
Training Iterations for each block of the parameter update unit	15
Training iterations for the auxiliary unit	10

4.7 Simulation Results

In this section, we study the performance of our proposed learning-based precoding schemes against the benchmark precoding techniques. We consider a single-cell MISO downlink in which the BS is equipped with four antennas ($N_t = 4$) that serve $K = 4$ single users. We generate 50,000 training and 2000 test samples of Rayleigh fading channel coefficients, respectively drawn from the same statistical distribution. The transmit data symbols are modulated using QPSK and 8PSK modulation schemes. The training SINR is randomly drawn from uniform distribution $\Gamma_{\text{train}} \sim \mathcal{U}(\Gamma_{\text{low}}, \Gamma_{\text{high}})$. Adam optimiser [130] is used for stochastic gradient descent algorithm with Lagrangian function as a loss metric.

Furthermore, a parametric rectified linear unit (**PReLu**) activation function is used for both convolutional and fully connected layers instead of the traditional **ReLu** function. The reason for this is to address the problem of dying gradient

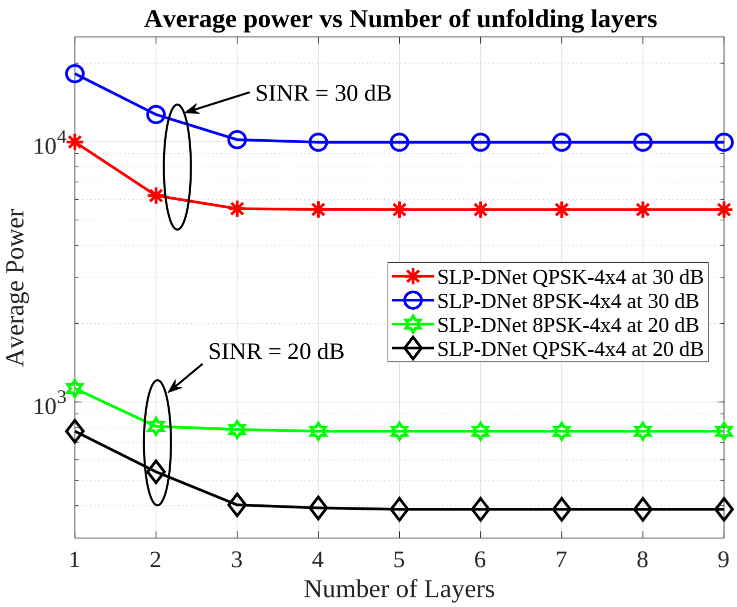
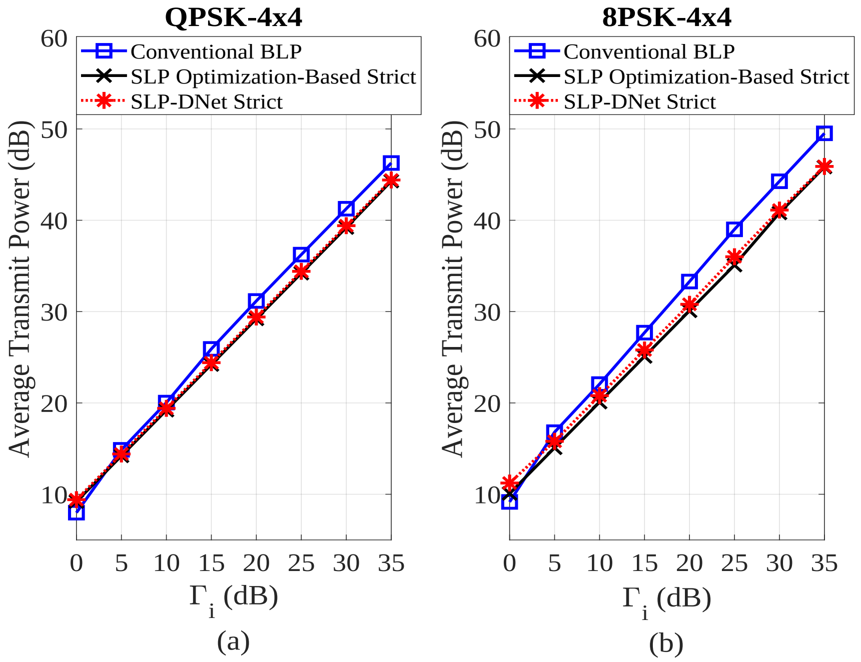


Figure 4.3: Transmit Power vs SINR averaged over 2000 test samples vs number of unfolding layers.

[130]. The learning rate is reduced by a factor $\alpha = 0.65$ after every iteration to aid the learning algorithm to converge faster. The learning models are implemented in Pytorch 1.7.1 and Python 3.7.8 on a computer with the following specifications: Intel(R) Core (TM) i7-6700 CPU Core, 32.0GB of RAM. Table 4.4 summarises the simulation parameters settings of the SLP-DNet. Generally, there has not been any standard rule for selecting the number of layers of deep neural network or deep unfolded network in theory and typical deep unfolding approaches select the numbers of layers in a heuristic manner [169]. Intuitively, to choose the appropriate number of layers in unfolding, we have run the experiments with the different number of unfolded blocks (layers) and plotted the average transmit power against the number of layers. Specifically, in our case, we find that the transmit power decreases with the number of layers until the power gains saturate beyond a certain number of layer, as shown in Fig. 4.3.

4.7.1 Performance of Non-Robust SLP-DNet

In this subsection, we evaluate the performance of our proposed unsupervised learning framework for nonrobust scenario against the benchmark algorithms [49, 73, 77] for both strict and relaxed angle rotations. Firstly, we compare the average trans-



(a)

(b)

Figure 4.4: Transmit Power vs SINR averaged over 2000 test samples for conventional BLP, SLP optimisation-based and nonrobust SLP-DNet schemes for $\mathcal{M}$-PSK modulation with $N_t = 4$, $K = 4$ under strict angle rotation.

mit power of the conventional BLP (2.22) described in subsection 2.5.1, the SLP optimisation-based problems (2.31), (4.9) and the SLP-DNet precoding scheme based on (4.29) and Algorithm 1. The performances of SLP-DNet and the benchmark schemes (conventional BLP and SLP optimisation-based) for strict angle rotation are shown in Figure 4.4. It can be observed that the transmit power of the proposed SLP-DNet closely matches the optimisation based SLP, both with significant gains against BLP.

Similarly, we discern the same trend in Figure 4.5 for the relaxed angle scenario as observed in Figure 4.4. Accordingly, we find from Figure 4.5 that the relaxed angle formulation offers significant power savings over the strict angle formulation and is therefore adopted in the subsequent experiments. Furthermore, at 30dB, the performance of SLP-DNet is within 5% of the SLP optimisation-based solution. Thus, while the SLP optimisation-based offers a slightly lower transmit power at SINR above 30dB, the proposed learning-based model's performance is within $96\% - 98\%$ of the optimisation-based solution.

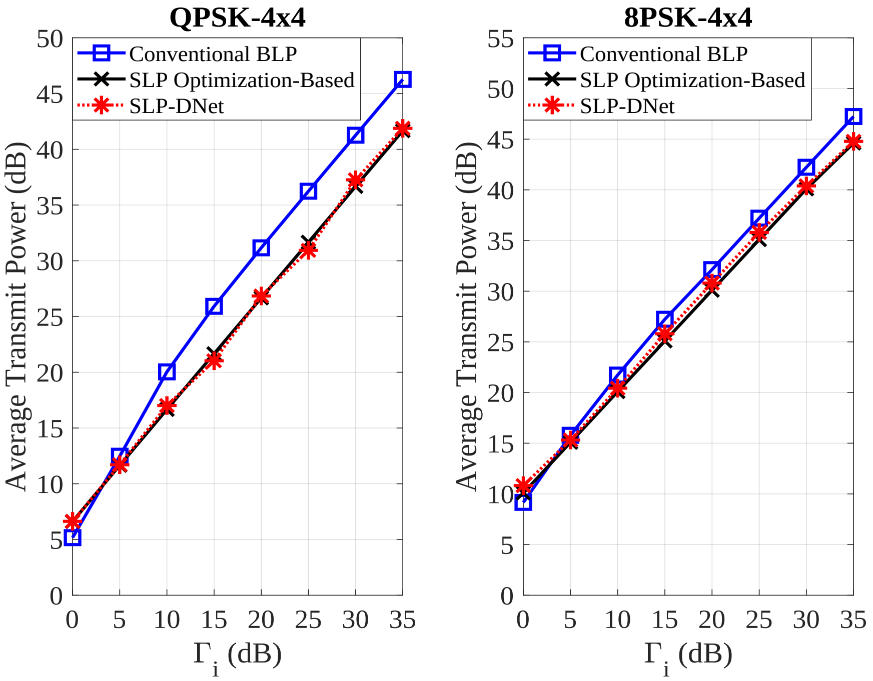


Figure 4.5: Transmit Power vs SINR averaged over 2000 test samples for conventional BLP, SLP optimisation-based and nonrobust SLP-DNet schemes for $\mathcal{M}$-PSK modulation with $N_t = 4$, $K = 4$ under relaxed angle rotation.

4.7.2 Performance of Robust SLP-DNet

In this subsection, we evaluate the performance of the robust SLP-DNet against the robust SLP optimisation-based and conventional precoding algorithms. We generate the results for the worst-case CSI error bounds between the range of $10^{-6} - 10^{-2}$.

Figures 4.6 and 4.7 compare the performance of the proposed robust SLP-DNet with the traditional robust block-level precoder [77] and robust SLP precoder [49] for the 4×4 MISO system evaluated at $\varsigma^2 = 10^{-4}$. For simplicity, we use QPSK modulation scheme. Figure 4.6 depicts how the average transmit power increases with the SINR thresholds, for CSI error bounds $\varsigma^2 = 10^{-4}$. The SLP optimisation-based precoding scheme is observed to show a significant power savings of more than 60% compared to the conventional BLP solution. Similarly, the proposed unsupervised learning-based precoder portrays a similar transmit power reduction trend. They show considerable power savings of $40\% - 58\%$ against the conventional BLP.

Furthermore, we investigate the effect of the CSI error bounds on the transmit

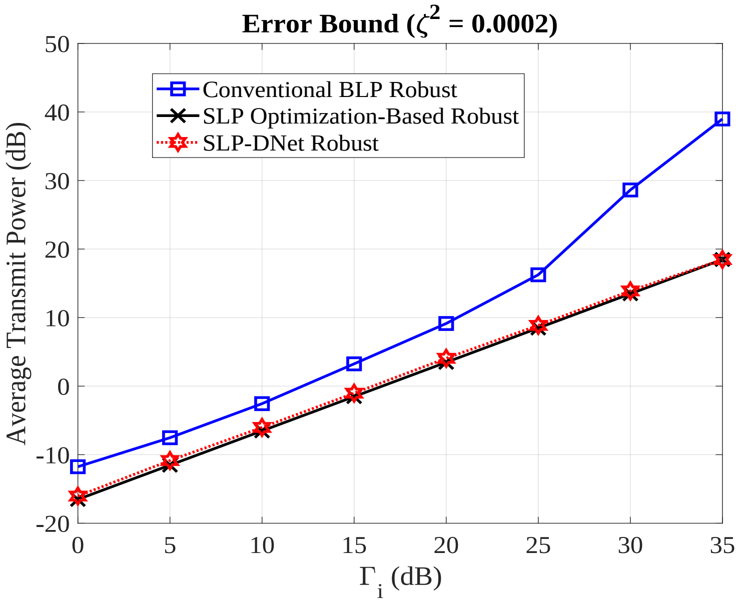


Figure 4.6: Transmit Power vs SINR averaged over 2000 test samples for robust conventional, SLP optimisation-based and SLP-DNet solutions with $N_t = 4$, $K = 4$ and $\varsigma^2 = 0.0002$.

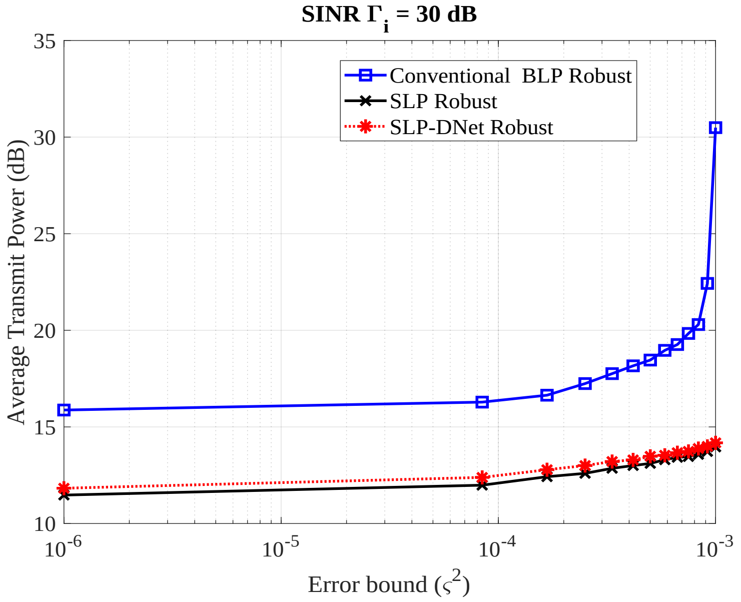


Figure 4.7: Transmit Power vs Error-bound for robust conventional BLP, SLP optimisation-based and SLP-DNet solutions with $N_t = 4$, $K = 4$.

power at 30dB SINR. Figure 4.7 depicts the transmit power variation with increasing CSI error bounds. Moreover, a significant increase in transmit power can be observed where the channel uncertainty lies within the region of CSI error bounds of $\varsigma^2 = 10^{-3}$. Interestingly, like the SLP optimisation-based algorithm, the pro-

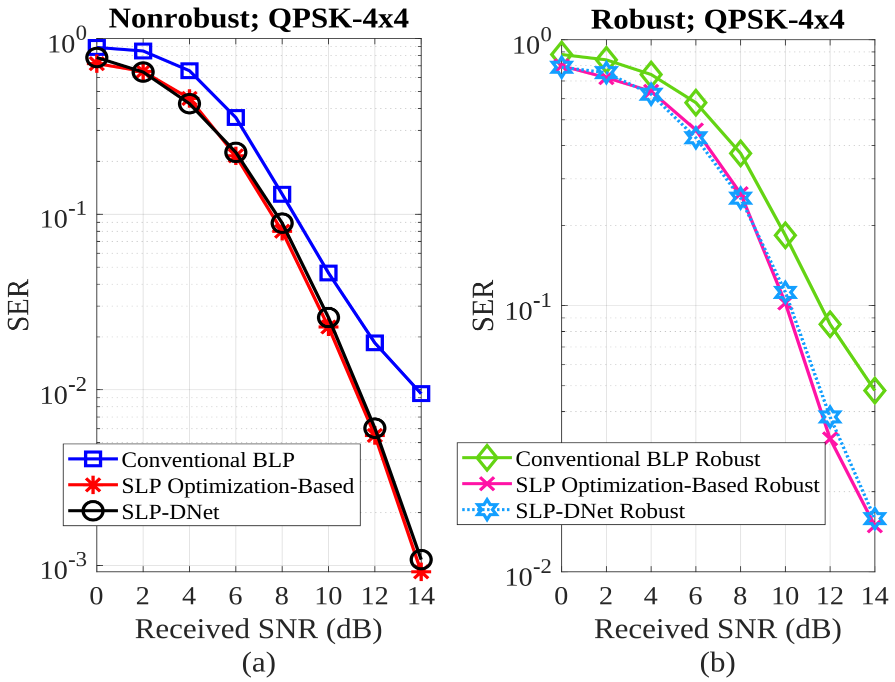


Figure 4.8: SER performance comparison of the conventional BLP, optimisation-based and SLP-DNet solutions with $N_t = 4$ and $K = 4$.

posed SLP-DNet also shows a descent or moderate increase in transmit power by exploiting the constructive interference.

4.7.3 Symbol-Error-Rate Evaluation

In this subsection, we evaluate the performance of the proposed SLP-DNet in terms of the received symbol error rate (SER) against the state-of-the-art precoding schemes. For a given SINR; $\Gamma_i = 12dB$, we first obtain the transmit powers for BLP and SLP optimisation-based schemes using (2.22) and (2.31), respectively. We repeat the same procedure for SLP-DNet using Algorithm 1. Given that the received $SNR = [0, \cdots, 14]$ in dB and the transmit power (P_T) obtained above, the noise spectral density (noise power) is calculated as flows: $N_0 = \frac{P_T}{SNR}$. Finally, the received symbol is: $y = \mathbf{h}^H \mathbf{w} s + \sqrt{N_0} \cdot n$.

Figure 4.8 (a) and Figure 4.8 (b) show the SER performances of the conventional BLP, the SLP optimisation-based schemes and the proposed SLP-DNet method for nonrobust and robust formulations. As expected, the SLP optimisation-based and the SLP-DNet outperforms BLP. We observe that SLP-DNet matches the SLP optimisation-based solution at lower SNR. As an illustration, we find that the

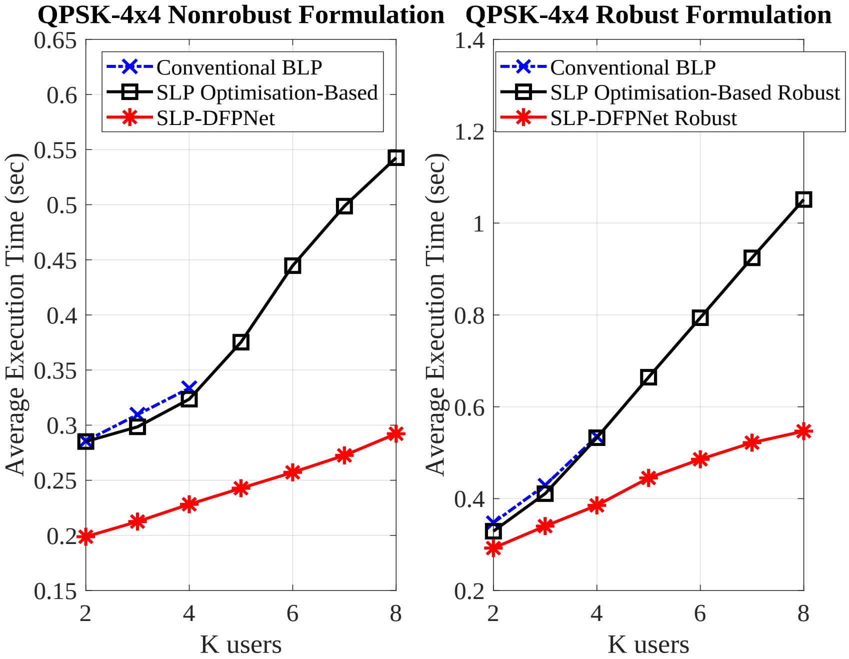


Figure 4.9: Comparison of average execution time per sample averaged over 200 test samples for conventional BLP, optimisation-based and SLP-DNet solutions with $N_t = 4$ and K users $(2, \cdots, 8)$.

performance gap between the SLP optimisation-based solution and the SLP-DNet at 10^{-2} SER is $0.0012dB$. This proves that with SLP via CI, interference from unintended users can be aligned constructively with symbols of interest at the receiver to improve received signal detection.

Figures 4.9(a) and 4.9(b) depict the execution times for nonrobust and robust formulations. It can be seen that both SLP optimisation-based algorithm and the proposed learning schemes produce solutions for N_t <K, while BLP fails.

Figure 4.9(a) shows the average execution time of the proposed unsupervised learning solutions per symbol averaged over 2000 test samples for nonrobust formulations. The SLP-DNet is observed to be significantly faster than the SLP optimisation-based. For example, the theoretical complexity is polynomial order-3 and polynomial order-6.5 or order-7.5 for SLP-DNet and conventional methods, respectively. This is shown in the execution times, where there is a significantly steeper increase in run-time as the number of users increases. The decrease in computational cost is because the dominant operations involved in SLP-DNet at the inference are simple matrix-matrix or vector-matrix convolution. The same trend is also observed in the case of a robust channel scenario, as shown in Figure 4.9(b). Therefore, the results in Figures 4.9(a) and 4.9(b) demonstrate that the proposed unsupervised learning-based precoding solutions offer a good trade-off between the performance and computational complexity. Moreover, as per the results obtained, SLP-DNet's performance is within the range of $89\% - 99\%$ of the optimal SLP optimisation-based precoding solution. Thus, our proposals demonstrate a favourable tradeoff between the performance and the computational complexity involved.

4.8 Summary

In this chapter, an unsupervised learning-based precoding schemes for a multi-user downlink MISO system have been proposed. The proposed learning technique exploits the CI for the power minimisation problem so that for given QoS constraints, the transmit power available for transmission is minimised. We use domain knowledge to design unsupervised learning architectures by unfolding the proximal IPM barrier *'log'* function derived from SLP power minimisation problem. The proposed learning scheme is then extended to robust precoding designs with imperfect CSI bounded by CSI errors. We demonstrate that our proposal is computationally efficient and allows for feasible solutions to be obtained for problems where traditional numerical optimisation like IPM and brute-force maximum likelihood solvers would not converge or would be prohibitively costly.

Chapter 5

Quantised DNN Frameworks for Symbol Level Precoding

5.1 Introduction

The previous chapter focuses on designing an unsupervised model-driven DL scheme for SLP (SLP-DNet), where the NN weights are computed using standard full precision floating point presentation (i.e., 32-bits). As seen in Chapter 4, the proposed learning strategy has a reduced complexity compared to the traditional optimisation solutions. This chapter will introduce a quantisation technique to further reduce the network's inference complexity at the device edge.

While CI-based precoding methods offer superior performance, computing them online on a symbol-by-symbol basis can be computationally demanding. To overcome this impediment, DL-based precoding designs that use domain knowledge have been recently proposed for MU-MISO downlink transmission [24, 28, 29]. However, the drawback of such methods is that the optimisation constraints are not directly integrated with the loss function. Moreover, their performance is bounded by the assumptions and accuracy of the optimal solutions obtained from the optimisation algorithm. To address these drawbacks, an unsupervised deep unfolding precoding design termed "SLP-DNet" that utilises the specifics of the optimisation objectives of the precoding problem described in Chapter 4 is adopted. We will use this model in this chapter as a benchmark to design its corresponding compressed versions, where the NN weights are quantised to lower numerical representations.

As explained in Chpater 3, subsection 3.4 that DL model contains thousands or even millions of learnable parameters, usually stored in a 32-bit floating-point (FP32) numerical presentation, making the model computationally and memory demanding during inference and deployment. To facilitate the online training and deployment of a trained DL model at the device edge, light-weight DNN designs with lower-precision numerical formats have gained significant attention within the deep learning community, typically applied to image processing applications [145, 170–172]. However, this concept has not been fully explored in wireless communications. In this chapter, we propose a DL model's structural simplification method through weights quantisation for learning-based SLP design. We adopt the SLP-DNet model. This chapter's contributions are summarised below:

- We propose a memory and complexity efficient DNN approach, applied to the learning-based precoding framework (SLP-DNet) described in Chapter 4. Specifically, we propose an efficient model simplification via weights compression to accelerate both training and inference to facilitate deployment on resource-constrained embedded hardware platforms.

- We devise a scalable tradeoff between performance and inference complexity, by allowing a percentage of the DNN weights to be quantised, while retaining important weights in full-precision. By tuning the percentage of quantised weights, a scalable tradeoff between performance and complexity / memory efficiency is achieved.

- We further introduce a stochastic quantisation (SQ) technique that uses the quantisation error to alleviate the loss in performance caused by the nonhomogenous quantisation errors of the conventional extreme quantisation (binary and ternary). In the SQ technique, a fraction of the NN weight matrix is quantised to lower resolution while the remaining is retained in its full-precision, resulting in a hybrid quantised weight matrix. The technique yields a memory-efficient DL-based SLP model with a good balance between the performance and the computational complexity.

5.2 System Model

In this section, we begin the system formulation by considering an MU-MISO downlink transmission in a single cell scenario where an N_t-antenna BS serves K single-antenna users.

By exploiting the multiuser interference through CI, we can design a precoding scheme that enhances the symbol detection by pushing the received signals away from the constellation detection boundaries without consuming extra transmission power [49]. We have seen in Chapter **4.2** that if the maximum angle shift in the CI region is zero, the interfering signals overlap entirely on the signal of interest ($\varphi = 0$), then the problem reduces to a strict phase angle optimisation (see Figure 2.4). It is important to note that the strict phase formulation is not appealing because it yields an increase in the transmission power compared to the corresponding relaxed version [85]. For this reason, we will concentrate on the relaxed angle formulation in this chapter. With reference to Figure 2.4 and the variables defined in Chapter **4.2** according to the description in [49], we further define the precoding and the channel matrices respectively for simplicity as $\hat{\mathbf{H}} = [\hat{\mathbf{h}}_1, \cdots, \hat{\mathbf{h}}_K]$ and $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_K]$. Therefore, the optimisation-based SLP for a nonrobust multicast power minimisation is given by

$$\begin{aligned} \min_{\{\mathbf{w_1}\}} \quad & \|\mathbf{w}_1\|_2^2 \\ \text{s.t.} \quad & \bar{a} \leq \Phi_i^T \Upsilon \mathbf{w}_1 \leq \bar{b}\ , \ \forall i. \end{aligned} \tag{5.1}$$

where $\Phi_i = \begin{bmatrix} \hat{\mathbf{h}}_{Ri} & \hat{\mathbf{h}}_{Ii} \end{bmatrix}^T$, $\mathbf{w}_1 = \begin{bmatrix} \mathbf{w}_R & -\mathbf{w}_I \end{bmatrix}^T$, $\Upsilon = \begin{bmatrix} \mathbf{O}_{N_t} & -\mathbf{I}_{N_t} \\ \mathbf{I}_{N_t} & \mathbf{O}_{N_t} \end{bmatrix} \in \mathbb{R}^{2N_t \times 2N_t}$. Similarly, we also define the following: $\bar{a} = -\left(\Phi_i^T \Upsilon \mathbf{w}_1 - \sqrt{\Gamma_i n_0}\right) \tan\phi$ and $\bar{b} = \left(\Phi_i^T \Upsilon \mathbf{w}_1 - \sqrt{\Gamma_i n_0}\right) \tan\phi$.

5.2.1 Learning-Based SLP Model (SLP-DNet)

In this subsection, we consider an unsupervised deep unfolding framework (SLP-DNet) that unfolds the IPM *'log'* barrier function based on the problem (5.1) by reformulating it as unconstrained subproblems per user as described in Chapter 4

$$\min_{\mathbf{w}\in\mathbb{R}^{\mathbf{2N_t}\times\mathbf{1}}} f(\mathbf{w}_1)+\upsilon B(\mathbf{w}_1), \tag{5.2}$$

where $B(\mathbf{w}_1)$ is the logarithmic barrier function and υ is the Lagrangian multiplier related to the inequality constraints. The learning architecture is based on an proximal *'log'* barrier IPM approach as described in Chapter 4, where the precoding vector for every l-th iteration is obtained from the following learning update rule

$$\mathbf{w}_1^{[l+1]} = \operatorname{prox}_{\gamma^{[l]}\upsilon^{[l]}B}\left(\mathbf{w}_1^{[l]} - \gamma^{[l]}\Delta\mathcal{K}(\mathbf{w}_1^{[l]},\lambda^{[l]})\right), \tag{5.3}$$

where $\mathcal{K}(\mathbf{w}_1^{[l]},\lambda^{[l]}) = \|\mathbf{w}_1\|_2^2+\lambda\mathbf{w}_1$, and $\Delta\left(\mathcal{K}(\mathbf{w}_1^{[l]},\lambda^{[l]}\right) = \frac{\partial\mathcal{K}(\mathbf{w}_1^{[l]},\lambda^{l})}{\partial\mathbf{w}_1^{[l]}}$. The parameter, λ is introduced as an additional constraint to provide more stability to the learning architecture. Intuitively, NN cascade layers can be formed from (5.3) as follows

$$\mathbf{w}_1^{[l+1]} = \operatorname{prox}_{\gamma^{[l]}\upsilon^{[l]}B}\left[\left(\mathbf{I}_{2N_t}-2\gamma^{[l]}\right)\mathbf{w}_1^{[r]}+\gamma^{[l]}\lambda^{[l]}\mathbf{1}^T\right], \tag{5.4}$$

where $\mathbf{1}\in\mathbb{R}^{1\times 2N_t}$ is a vector of ones. By letting $\mathbf{W}_l = \mathbf{I}_{2N_t}-2\gamma^{[l]}$, $\mathbf{b}_l = \gamma^{[l]}\lambda^{[l]}\mathbf{1}^T$ and $\Theta_l = \operatorname{prox}_{\gamma^{[l]}\upsilon^{[l]}B}$, the l-layer network $\mathcal{L}^{[l-1]}\cdots\mathcal{L}^{[0]}$ will correspond to the l-th layers NN as described by (4.32) in Chapter 4. The nonlinear activation functions are defined by Θ_l. Based on the above formulations, it can be recalled that the SLP-DNet architecture has two main units; the parameter update module (PUM) and the auxiliary processing module (APM). The PUM has three core components associated with Lagrangian multiplier (υ), the auxiliary parameter (λ), and the training step-size (γ), which are updated based on the following

$$\mathcal{D}(\mathbf{w}_1,\upsilon,\gamma,\lambda) = \operatorname{prox}_{\gamma^{[l]}\upsilon^{[l]}B}\left(\mathbf{w}_1^{[l]} - \gamma^{[l]}\Delta\mathcal{K}(\mathbf{w}_1^{[l]},\lambda^{[l]})\right). \tag{5.5}$$

The structure that is related to the inequality constraint in (5.1) is the proximity barrier term. It is constructed with one convolutional layer, an average pooling layer, a fully connected layer, and a softPlus layer to constrain the output to a positive real value to satisfy the inequality constraint. The loss function over N batch training samples (batch size or the number of channel realisation) is Lagrangian function expressed as

$$\begin{aligned}\mathcal{L}(\mathbf{w}_1, \upsilon_1,\ \upsilon_2) = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{w}_1\|_2^2 + \frac{1}{N}\sum_{i=1}^{N}\left(\upsilon_1\left(\Phi_i^T\Upsilon\mathbf{w}_1 - \Phi_i^T\mathbf{w}_1\tan\phi + \sqrt{\Gamma_i n_0}\right)\right)\\ -\frac{1}{N}\sum_{i=1}^{N}\left(\upsilon_2\left(\Phi_i^T\Upsilon\mathbf{w}_1 + \Phi_i^T\mathbf{w}_1\tan\phi - \sqrt{\Gamma_i n_0}\right)\right) + \frac{\mu}{NL}\sum_{i=1}^{N}\sum_{l=1}^{L}\|\Omega_i\|_2^2,\end{aligned} \tag{5.6}$$

where Ω_i are the trainable parameters of the l-th layers associated with the weights and biases, and $\mu > 0$ is the penalty parameter that controls the bias and variance of the trainable coefficients.

The optimal precoder is obtained from the Lagrangian function (5.6) as

$$\mathbf{w}_1 = \frac{\left(\upsilon_1^T + \upsilon_2^T\right)\cdot\Phi_i\tan\phi - \left(\upsilon_1^T - \upsilon_2^T\right)\cdot\Upsilon^T\Phi_i\tan\phi}{2}. \tag{5.7}$$

5.2.2 Robust SLP-DNet

In a similar fashion to the above, we can derive a CSI-robust SLP-DNet from the robust SLP formulation under worst-case CSI-error. The robust SLP is given by [173]

$$\begin{aligned}\min_{\{\mathbf{w}_2\}}\quad & \|\mathbf{w}_2\|_2^2\\ \text{s.t.}\quad & \Phi^T\mathbf{Q}_1\mathbf{w}_2 + \varsigma\|\mathbf{Q}_1\mathbf{w}_2\|_2 + \sqrt{\Gamma n_0}\tan\phi \le 0\ \forall i\\ & \Phi^T\mathbf{Q}_2\mathbf{w}_2 + \varsigma\|\mathbf{Q}_2\mathbf{w}_2\|_2 + \sqrt{\Gamma n_0}\tan\phi \le 0\ \forall i.\end{aligned} \tag{5.8}$$

For convenience, we introduce new notations as follows: $\mathbf{Q}_1 = (\Upsilon - \mathbf{I}_{2N_t}\tan\phi)$ and $\mathbf{Q}_2 = (\Upsilon + \mathbf{I}_{2N_t}\tan\phi)$ and ς^2 is the CSI error bound. (5.8) is a second order cone programming (SOCP) and can be solved using convex optimisation software package.

It is important to note that the structure of the robust SLP-DNet is obtained by following similar steps from (5.2)-(5.5) by transforming (5.8) to its equivalent unfolded IPM *'log'* barrier form. The loss function is obtained from the Lagrangian of (5.8) as

$$\begin{aligned}\mathcal{L}_{\text{robust}}(\mathbf{w}_2,\ \upsilon_1,\ \upsilon_2) = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{w}_2\|_2^2 \\ + \frac{\upsilon_1}{N}\sum_{i=1}^{N}\left(\varsigma^2\|\mathbf{Q}_1\mathbf{w}_2\|_2^2 - \left(\sqrt{\Gamma n_0}\tan\phi - \Phi^T\mathbf{Q}_1\mathbf{w}_2\right)^2\right) \\ + \frac{\upsilon_2}{N}\sum_{i=1}^{N}\left(\varsigma^2\|\mathbf{Q}_2\mathbf{w}_2\|_2^2 - \left(\sqrt{\Gamma n_0}\tan\phi - \Phi^T\mathbf{Q}_2\mathbf{w}_2\right)^2\right) + \frac{\mu}{NL}\sum_{i=1}^{N}\sum_{i=1}^{L}\|\Omega_i\|_2^2. \quad (5.9)\end{aligned}$$

where $\begin{bmatrix}\|\mathbf{Q}_1\|_2^2 & \|\mathbf{Q}_2\|_2^2\end{bmatrix} = \tilde{\mathbf{Q}}_{\text{norm}}$, $\begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix} = \tilde{\mathbf{Q}}$ and $\begin{bmatrix}\upsilon_1 & \upsilon_2\end{bmatrix} = \tilde{\upsilon}$.

The optimal precoder can be easily obtained from (5.9)

$$\mathbf{w}_2 = -\Phi\tilde{\mathbf{Q}}\tilde{\upsilon}^T\mathbf{X}^{-1}\sqrt{\Gamma n_0}\tan\phi, \quad (5.10)$$

where $\mathbf{X} = \left(\mathbf{I}_{2N_t} + \tilde{\mathbf{Q}}_{\text{norm}}\tilde{\upsilon}^T\left(\varsigma^2 - \Phi^T\Phi\right)\right)$. Note that the Lagrange multipliers υ_1 and υ_2 are associated with the barrier term and are randomly initialised from a uniform distribution.

The architecture of SLP-DNet is contingent upon the above formulations, as depicted in Figure 4.1 of hapter 4. This is also similar to the structure of the robust SLP-DNet (RSLP-DNet). However, contrary to SLP-DNet, the input optimal precoding tensor for the robust SLP-DNet is initialised using (5.10), which forms the Lagrangian module as shown in Figure 5.1. For clarity, we summarise the architectures of the APM and the barrier term we use our proposed designs in Tables while Tables 5.1 and 5.2.

5.3 NN Weight Quantisation

Several DNNs compression techniques have been proposed for efficient edge inference to tackle the ever-increasing model size problem. The ultimate criterion of such methods is that lower inference computation and memory efficiency overheads can be achieved with minimal accuracy loss. Typically, the weights of l-th layer DNN architecture are represented by [170] $\mathcal{W}_l = \{\mathbf{W}_i, \cdots, \mathbf{W}_m\}$ for $\forall\, i = 1, \cdots, m$,

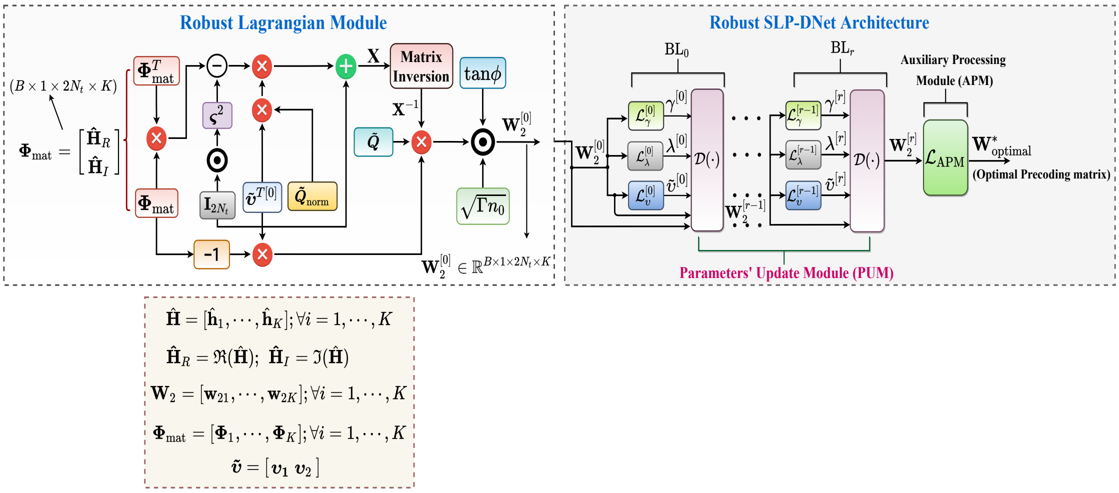

Robust Lagrangian Module
$(B \times 1 \times 2N_t \times K)$
$\mathbf{\Phi}_{\text{mat}} = \begin{bmatrix} \hat{\mathbf{H}}_R \\ \hat{\mathbf{H}}_I \end{bmatrix}$
$\mathbf{\Phi}^T_{\text{mat}}$
$\mathbf{\Phi}_{\text{mat}}$
ς^2
$\mathbf{I}_{2N_t}$
$\tilde{\boldsymbol{v}}^{T[0]}$
$\tilde{Q}_{\text{norm}}$
$\mathbf{X}$
Matrix Inversion
$\mathbf{X}^{-1}$
$\tan\phi$
$\tilde{Q}$
$\mathbf{W}_2^{[0]}$
$\sqrt{\Gamma n_0}$
-1
$\mathbf{W}_2^{[0]} \in \mathbb{R}^{B \times 1 \times 2N_t \times K}$
Robust SLP-DNet Architecture
BL_0
BL_r
Auxiliary Processing Module (APM)
$\mathbf{W}_2^{[0]}$
$\mathcal{L}_\gamma^{[0]}$
$\mathcal{L}_\lambda^{[0]}$
$\mathcal{L}_v^{[0]}$
$\gamma^{[0]}$
$\lambda^{[0]}$
$\tilde{v}^{[0]}$
$\mathcal{D}(\cdot)$
$\mathbf{W}_2^{[r-1]}$
$\mathcal{L}_\gamma^{[r-1]}$
$\mathcal{L}_\lambda^{[r-1]}$
$\mathcal{L}_v^{[r-1]}$
$\gamma^{[r]}$
$\lambda^{[r]}$
$\tilde{v}^{[r]}$
$\mathbf{W}_2^{[r]}$
$\mathcal{L}_{\text{APM}}$
$\mathbf{W}^*_{\text{optimal}}$
(Optimal Precoding matrix)
Parameters' Update Module (PUM)
$\hat{\mathbf{H}} = [\hat{\mathbf{h}}_1, \cdots, \hat{\mathbf{h}}_K]; \forall i = 1, \cdots, K$
$\hat{\mathbf{H}}_R = \Re(\hat{\mathbf{H}}); \ \hat{\mathbf{H}}_I = \Im(\hat{\mathbf{H}})$
$\mathbf{W}_2 = [\mathbf{w}_{21}, \cdots, \mathbf{w}_{2K}]; \forall i = 1, \cdots, K$
$\mathbf{\Phi}_{\text{mat}} = [\mathbf{\Phi}_1, \cdots, \mathbf{\Phi}_K]; \forall i = 1, \cdots, K$
$\tilde{\boldsymbol{v}} = [\boldsymbol{v}_1 \ \boldsymbol{v}_2]$

Figure 5.1: Learning-based robust symbol level precoding (RSLP-DNet) Architecture

Table 5.1: An APM NN Architecture

Layer	Parameter, kernel size $= 3 \times 3$
Input Layer	Input size $(\text{B},\ 1,\ 2N_t,\ K)$
Layer 1: Convolutional	Size $(\text{B},\ 16,\ 2N_t,\ K)$, dilation $= 1$ and unit padding
Layer 2: Batch Normalisation	eps $= 10^{-6}$, momentum $= 0.1$
Layer 3: Activation	PReLu/k-bit function
Layer 4: Convolutional	Size $(\text{B},\ 8,\ K,\ 2KN_t)$, dilation $= 1$ and unit padding
Layer 5: Batch Normalisation	eps $= 10^{-6}$, momentum $= 0.1$
Layer 6: Activation	PReLu (k-bit function)
Layer 7: Convolutional	Size $(\text{B},\ 1,\ 2KN_t,\ 1)$, dilation $= 1$ and unit padding

Table 5.2: Proximity Barrier Term NN Architecture

Layer	Parameter, kernel size $= 3 \times 3$
Input Layer	Input size $(\text{B},\ 1,\ 2N_t,\ K)$
Layer 1: Convolutional	Size $(\text{B}, 20, 2N_t, K^2)$; zero padding
Layer 2: Average Pooling	Size $((1,\ 1),\ \text{stride} = (1,\ 1))$
Layer 3: Activation	Soft-Plus
Layer 4: Flat	Size $(\text{B} \times 40 \times K^2)$
Layer : Fully-connected	Size$(\text{B} \times 40 \times K^2,\ 1)$
Layer 5: Activation	Soft-Plus function

where m is the number of kernels/filters (output channels). The n-dimensional weight tensor $\mathbf{W}_i \in \mathbb{R}^n$, $n = c \times w \times h$ in l-th convolutional layer, where $c \times w \times h$ represents the input channels, filter width and filter height respectively, and for a fully connected layer, $n = m \times c$ (number of the output and input neurons, respectively). For convenience, in what follows, we drop the kernel subscript.

5.3.1 Binary Weights

The real-valued weights are converted to $(\mathbf{B}_w \in \{+1, -1\}^n)$. A full-precision 32-bit weight matrix is binarised as follows[174]

$$\mathbf{B}_w = sign(\mathbf{W}) = \begin{cases} +1 & \text{if } \mathbf{W} \geq 0 \\ -1 & \text{otherwise,} \end{cases} \tag{5.11}$$

A more robust binarised weight "BWN" is proposed as an extension of a straightforward binary network (Binary Connect) by introducing a real scaling factor $\beta \in \mathbb{R}^+$ such that $\mathbf{W} \approx \beta \mathbf{B}_\mathrm{w}$ by solving an optimisation problem [170]

$$J(\mathbf{B}_\mathrm{w}, \beta) = \underset{(\mathbf{B}_\mathrm{w}, \beta)}{\mathrm{argmin}} \quad \|\mathbf{W} - \beta \mathbf{B}_\mathrm{w}\|_2^2, \tag{5.12}$$

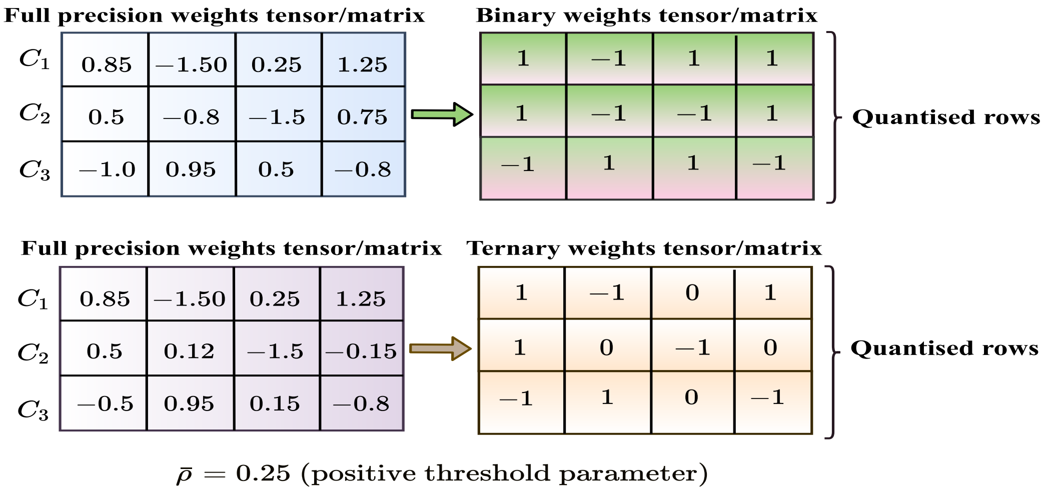
and this yields

$$\begin{aligned} \mathbf{B}_\mathrm{w}^* &= sign(\mathbf{W}) \\ \beta^* &= \frac{1}{n} \|\mathbf{W}\|_1 \end{aligned} \tag{5.13}$$

5.3.2 Ternary Weights

A ternary weighted network (TWN) is the one in which an extra 0 state is introduced into BWN to solve the following optimisation problem [175]

$$\begin{cases} \beta^*, \mathbf{B}_\mathbf{W}^* = & \underset{\beta,\ \mathbf{B}_\mathrm{w}}{\mathrm{argmin}}\, J(\beta,\ \mathbf{B}_\mathrm{w}\ ) = \|\mathbf{W} - \beta \mathbf{B}_\mathrm{w}\|_2^2 \\ \text{s.t.} & \beta \geq 0,\ \mathbf{B}_\mathrm{w} \in \{-1,\ 0,\ +1\}^n, \end{cases} \tag{5.14}$$


Full precision weights tensor/matrix
Binary weights tensor/matrix
Quantised rows
Full precision weights tensor/matrix
Ternary weights tensor/matrix
Quantised rows
$\bar{\rho} = 0.25$ (positive threshold parameter)

Full precision weights tensor/matrix

C_1	0.85	−1.50	0.25	1.25
C_2	0.5	−0.8	−1.5	0.75
C_3	−1.0	0.95	0.5	−0.8

Binary weights tensor/matrix

1	−1	1	1
1	−1	−1	1
−1	1	1	−1

Full precision weights tensor/matrix

C_1	0.85	−1.50	0.25	1.25
C_2	0.5	0.12	−1.5	−0.15
C_3	−0.5	0.95	0.15	−0.8

Ternary weights tensor/matrix

1	−1	0	1
1	0	−1	0
−1	1	0	−1

Figure 5.2: Binary and Ternary DNN weights

and solving (5.14) gives

$$\mathbf{B}_{\mathrm{w}}^{*} = \begin{cases} +1 & \text{, if } \mathbf{W} > \bar{\rho} \\ 0 & \text{, if } |\mathbf{W}| \leq \bar{\rho} \\ -1 & \text{, if } \mathbf{W} < -\bar{\rho}, \end{cases} \tag{5.15}$$

where $\bar{\rho} = \frac{0.7}{n} \sum\limits_{i=1}^{n} |\mathbf{W}|$ and $\beta^{*} = \frac{1}{\mathbf{I}_{\bar{\rho}}} \sum\limits_{i \in \mathbf{I}_{\bar{\rho}}} |\mathbf{W}|$,
$\mathbf{I}_{\bar{\rho}} = \{|\mathbf{W}| > \bar{\rho}\}$ is the cardinality of set $\mathbf{I}_{\bar{\rho}}$. As an illustration, Figure 5.2 depicts how the weight matrices are quantised predicated on (5.13) and (5.15), respectively.

5.4 Proposed Quantised SLP-DNet Design

This section introduces the DNN model compression technique via NN weight stochastic quantisation, where numerical values are reduced to lower precision in parts based on the quantisation errors due to extreme quantisations (binary and ternary).

5.4.1 NN Weights Quantisation and Stochastic Division

The existing works on low-bit DNN design focus only on reducing the bit-widths of the weights and activations to speed up the training and inference times and also improve memory efficiency. However, in low-bit DNN designs, the impact of quantisation on the performance of the learning algorithm has not been fully explored and understood. In this subsection, the quantisation introduced termed stochastic quantisation (SQ) is done using a linear probability function for selecting the filter weights to be quantised for designing a low-bit scalable learning-based precoder.

The weight matrix of each layer of the DNN can be expressed as: $\mathcal{W} = \{\mathbf{W}_1, \cdots, \mathbf{W}_n\}$. Here, the rows of the weight matrix are partitioned into two parts according to the following

$$\mathcal{W} = \{\mathcal{W}_q, \mathcal{W}_f\}, \tag{5.16}$$

where $\mathcal{W}_q = \{\mathbf{W}_{q1}, \cdots, \mathbf{W}_{qM}\}$ and $\mathcal{W}_f = \{\mathbf{W}_{f1}, \cdots, \mathbf{W}_{fN}\}$ represent the quantised and full-precision parts of the weight respectively, and should satisfy the condition below

$$\mathcal{W} = \mathcal{W}_q \cup \mathcal{W}_f \text{ and } \mathcal{W}_q \cap \mathcal{W}_f = \emptyset. \tag{5.17}$$

As seen from (5.16), one subset of the weight $\mathcal{W}_q$ is quantised to a low bit-width while the remaining $\mathcal{W}_f$ is kept in its full-precision form, so that the entire weights matrix is composed of both binary and floating-point values. Note that a fully quantised DNN can be obtained by setting $\mathcal{W}_f$ to a null set.

Suppose r_{sq} is the quantisation ratio (QR) (i.e., the percentage of weights quantised as a fraction of the total weights in the DNN), and n is the length of the weight matrix (number of elements), the number of elements in the quantisation group is $M_q = r_{sq}n$ while that of a full-precision parts is $M_f = (1 - r_{sq})n$. The QR can be gradually increased to 100% until the entire network is finally quantised. To select the channel to be quantised, we use the lottery disc algorithm shown in Figure 5.3. It can be observed in Figure 5.3 that each sector of the disc represents a probability of selecting a channel (row of weight matrix). The disc is rotated by choosing a value from the uniform distribution whose magnitude is slightly above the probability value. After every selection, the probability is reset (i.e., $pr_j = 0$) to ensure that a channel is selected without replacement as summarised in Algorithm 2.

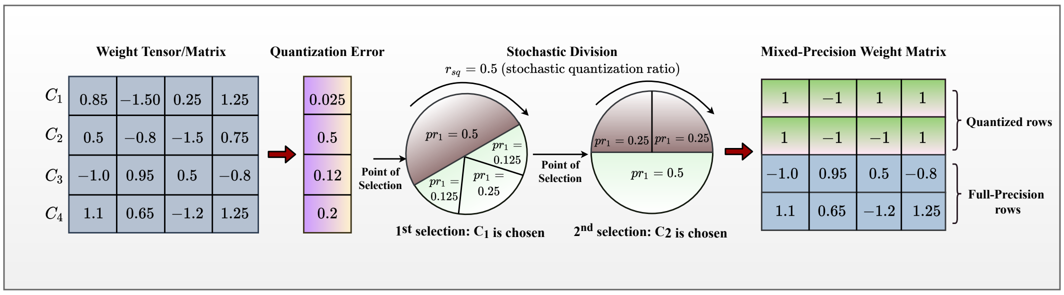


Figure 5.3: Stochastic Quantisation Weight Matrix Partitioning Procedure

Algorithm 2 Circular Lottery Algorithm for Weight matrix Division

```
Input: $r_{sq}$ Stochastic Quantisation ratio and Weight matrix ($\mathcal{W}$)
Output: $\mathcal{W}_q$ and $\mathcal{W}_f$
Initialisation:
$\mathcal{W}_q = \mathcal{W}_r = \emptyset$
Compute QP function $\mathbf{pr} \in \mathbb{R}^n \forall i \; \{i = 1, \cdots, n\}$ based on (5.18)
$M_q = r_{sq} n$
for $j = 1$ to $M_q$ do
    $\hat{\mathbf{pr}} = \frac{\mathbf{pr}}{\|\mathbf{pr}\|_1}$ (normalised probability)
    Select a random value $\vartheta_j \in \{0, 1\}$ from a random uniform distribution
    Set $s_j = 0$ and $i = 0$
    while $s_j < \vartheta_j$ do
        $i = i + 1$
        $s_j = s_j + \hat{pr}_j$
    end while
    Compute: $\mathcal{W}_q = \mathcal{W}_q \cup \mathcal{W}$
    Reset $pr_i = 0$ {This is to avoid i-th channel weight from being selected again}
end for$\mathcal{W}_r = \mathcal{W} \setminus \mathcal{W}_q$
```

5.4.2 Quantisation Error and Quantisation Probability

Recall that classical binarized DNNs suffer a significant performance loss due heterogeneous nature of the quantisation error (QE) over the entire network. The performance can, however, be improved by stochastically selecting the filter or channel weight matrix to be quantised using a random probability distribution based on the QE between the real-valued and quantised weights as follows

$$e_j = \frac{\|\mathbf{W}_j - \mathbf{Q}_j^*\|_2}{\|\mathbf{W}_j\|_2}; \tag{5.18}$$

where $\mathbf{Q}_j^*$ could be binary or ternary based on (5.13) or (5.15).

We define the vector of the *n-th* row weight matrix of a given layer as

$\mathbf{e} = [e_1, \cdots, e_n]$. The quantisation probability is formulated such that a higher probability is assigned to filter/weights if the quantisation error is small because quantising these weights does not yield a significant loss of accuracy or performance. For a given weight matrix, QR, and quantisation probability (QP), a channel is randomly sampled without replacement using a circular lottery Algorithm 2. From this, we can observe that the QP function is inversely proportional to QE and is defined as $f_p = \frac{1}{e+\delta}$, where $\delta = 10^{-6}$ to avoid possible numerical overflow. The QP function is monotonically non-decreasing to prioritise the selection of the channels/weights to be quantised. Different monotonically non-decreasing functions are:

- Uniform function: $pr_j = \frac{1}{n}$, n is the number of the neurons or length of the rows of each layer weight matrix.
- Linear function: $pr_j = \frac{f_{p_j}}{\sum_j f_{p_j}}$
- Half-Gaussian function: $pr_j = \frac{\sqrt{2}}{\sigma\sqrt{\pi}}\exp\left(\frac{-f_{p_j}^2}{2\sigma^2}\right)$
- Softmax function: $pr_j = \frac{\exp(f_{p_j})}{\sum_j \exp(f_{p_j})}$

The simplest of these QP functions is uniform or constant function but is not appealing because it is independent of the QE and therefore ignores the random quantisation proposition. The most intriguing of all is the half-Gaussian function because of the extra parameter (σ), which can be learned but is more complicated. The linear and softmax functions have been found to yield nearly the same performance, but the former is simpler to implement. Accordingly, in this work, we use the linear function because it balances between performance and simplicity.

5.4.3 Low-bit Activation Function

The inputs to convolutional and fully connected layers are the outputs of the previous layers' activations. In many low-bit DNN designs, the activation layer is often left in its full-precision. However, quantising the activation layer is crucial in replacing the floating-point operations with more efficient binarisation. The conventional activation functions such as *"Relu"* may not be suitable for low-bit DNN [176].

Therefore, the activations are quantised from 32-bit ($\mathbf{W}_{32}$) to $k-$bit according to the function

$$\mathbf{W}_b = \frac{round\left((\mathbf{W}_{32} - x) \cdot \left(2^k - 1\right) / (y - x)\right)}{\left(2^k - 1\right)} \tag{5.19}$$

where $\mathbf{W}_{32}$ is the floating-point activation bounded by the input dimension (x, y) and $k = 2$. The activations are not stochastically quantised because, unlike in weights, the activations do not have learning parameters.

5.5 Model training and Inference

The fully quantised versions of SLP-DNet based on binary and ternary bits (SLP-DBNet and SLP-DTNet) are trained the same way as plane SLP-DNet. However, back-propagation through the quantisation function results in zero gradients due to the thresholding that summarises the activations or outputs into binary values. This lack of gradient results in the network not learning anything. A straight-through estimator (STE) [177] is used in the backward pass to solve this problem. Specifically, STE bypasses the derivative of the threshold function and passes on the incoming gradient as if the function was an identity function expressed as [177]

$$clip(x, -1, 1) = \max(-1, \min(1, x)). \tag{5.20}$$

In the PUM, each block contains three main components and is trained blockwise for k-th number of iterations as explained in Chapter 4. Similarly, APM is trained for r-th iterations, and the number of training iterations of the PUM and APM may not necessarily be equal. The PUM is trained for 20 iterations and the APM for 10 iterations. We modify the learning rate by a factor $\alpha \in \mathbb{R}^+$ for every training step to improve the training efficiency using a stochastic gradient descent algorithm with Adam optimiser [130].

5.5.1 Stochastically quantised SLP-DNet (SLP-DSQNet)

The SLP-DSQNet training is slightly different from that of SLP-DNet. The training is summarised in four stages: stochastic weight matrix division, forward propagation, backward propagation, and parameter update. Given QR, the weight matrix is partitioned into a quantisation group and a full-precision group using Algorithm 2. A hybrid weight is then formed containing the quantised and the real-valued weights, and it provides a better gradient direction than pure quantised weights. If $\tilde{\mathcal{W}}_{qf}$ is the composite weight matrix, the weight update with respect to the composite gradients is given by $\mathcal{W}^{r+1} = \mathcal{W}^{r} - \eta \frac{\partial \mathcal{L}}{\partial \tilde{\mathcal{W}}^{r}_{qf}}$. We train the network with different QRs, which are fixed for all the training iterations and inference.

The learning is performed in an unsupervised fashion in which the loss function is the Lagrangian function's statistical mean over the training batch. During the inference, a feed-forward pass is performed over the whole layers using the learned Lagrangian multipliers to compute the precoding vector using (5.7) and (5.10) for nonrobust and robust SLP formulations. Note that except where necessary stated, the training SINR is drawn from a random uniform distribution to enable learning across a wide range of SINR values.

5.6 Computational Complexity Analysis

This section presents the analytical evaluations of the computational costs of the proposed SLP-DSQNet precoding schemes and compares them with SLP-DNet, the conventional BLP, and the SLP optimisation-based methods. The complexities are computed in terms of the number of real arithmetic operations involved. To derive the analytical complexity of the optimisation-based SLP, we first convert the second-order cone programming (SOCP) (5.1) into standard linear programming (LP) as follows

$$\begin{aligned} \min_{\{\mathbf{w}_1\}} \quad & \|\mathbf{w}_1\|_2^2 \\ \text{s.t.} \quad & |\Phi_i^T \Upsilon \mathbf{w}_1| \leq \bar{b} \ , \ \forall i. \end{aligned} \tag{5.21}$$

where $\bar{b} = \left(\Phi_i^T \Upsilon \mathbf{w}_1 - \sqrt{\Gamma_i n_0}\right) \tan\phi$. To convert (5.21) to its equivalent LP, we introduce new optimisation variable

$$\begin{aligned} \min_{\{\mathbf{x}\}} \quad & \mathbf{d}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{d}_k^T \mathbf{x} \leq -\tan\phi \sqrt{\Gamma_i n_0} \ , \ \forall i \end{aligned} \tag{5.22}$$

where $\mathbf{d} = [0 \quad \mathbf{w}_1^T]^T \in \mathbb{R}^{(2N_t+1)\times 1}$, $\mathbf{x} = [1 \;\; \mathbf{w}_1]^T \in \mathbb{R}^{(2N_t+1)\times 1}$, and $\mathbf{d}_k = \left[\left|\Phi_i^T \Upsilon \mathbf{w}_1\right| - \Phi_i^T \tan\phi\right]^T \in \mathbb{R}^{(2N_t+1)\times 1}$.

Given the optimal target accuracy, $\varepsilon > 0$, the complexity of solving convex optimisation via IPM is characterised by the formation (C_{form}) and factorisation (C_{fact}) of the matrix coefficients with $\bar{n}$ linear equations having $\bar{n}$ unknowns and is given by [168]

$$C_{\text{total}} = (C_{\text{form}} + C_{\text{fact}}) \times \ln\left(\frac{1}{\varepsilon}\right)\sqrt{\sum_{j=1}^{M_{\text{lc}}} Q_j + 2M_{\text{sc}}} \tag{5.23}$$

where Q represents the constraint's dimension, M_{lc} and M_{sc} denote the numbers of linear inequality matrix and second order cone (SOC) constraints, respectively. Therefore, the overall complexity is

$$C_{\text{total}} = \underbrace{\left[\underbrace{\bar{n}\sum_{j=1}^{M_{\text{lc}}} Q_j^3 + \bar{n}^2 \sum_{j=1}^{M_{\text{lc}}} Q_j^2}_{\text{due to } M_{\text{lc}}} + \underbrace{\bar{n}\sum_{j=1}^{M_{\text{sc}}} Q_{j=1}^2}_{\text{due to } M_{\text{sc}}} + \bar{n}^3\right]}_{C_{\text{form}}+C_{\text{fact}}} \times \ln\left(\frac{1}{\varepsilon}\right)\sqrt{\sum_{j=1}^{M_{\text{lc}}} Q_j + 2M_{\text{sc}}}. \tag{5.24}$$

It can be observed that (5.22) has K constraints with dimension $2N_t + 1$. Therefore, using (5.24), the total computational cost is obtained as

$$C_{\text{total}} = \sqrt{2N_t + 1}\left[\bar{n}(2N_t + 1) + \bar{n}(2N_t + 1)^2 + \bar{n}^3\right]\ln\left(\tfrac{1}{\varepsilon}\right). \tag{5.25}$$

By following similar principles and steps above, we can obtain the complexities of the robust SLP and the conventional BLP schemes.

On the other hand, to determine the complexities of our proposed precoders, we first evaluate the complexities of the learning modules (PUM and APM) in terms of arithmetic operations involved. For APM, there are three convolution blocks. The feature map determines the arithmetic operations for a convolution layer and is given by the number of multiplications and additions involved in the convolution operation. The number of operations in a given convolutional layer is

$$C_{\text{conv}} = \left(c_{\text{in}}k_{\text{f}}^{2} + (c_{\text{in}}k_{\text{f}}^{2} - 1) + 1\right) c_{out} N_{\text{w}} N_{\text{h}} \tag{5.26}$$

where N_{h}, N_{w}, k_{f}, C_{in} and C_{out} denote the height, width of the input layer tensor, filter size, number of input and output channels, respectively. It is important to note that only the first and second convolutions are quantised, while the last convolution is not to avoid losing essential features of the output precoder. Since in our proposed approach, the layer weight matrix contains both floating points and quantised entries, then the quantisation approximation of the convolution has $\frac{1}{32}\left(c_{\text{in}}k_{\text{f}}^{2}N_{\text{w}}N_{\text{h}}c_{out}\right) \times QR$ binary operations and $\left(c_{\text{in}}k_{\text{f}}^{2}N_{\text{w}}N_{\text{h}}c_{out}\right) \times (1-QR)$ non binary operations based on (5.26). Using these expressions, we obtain the generic complexity of the APM as

$$\begin{aligned} C_{\text{APM}} = & \underbrace{\frac{1}{32}\sum_{l=1}^{L} N_{\text{h}}^{[l-1]} N_{\text{w}}^{[l-1]} \left[C_{\text{in}}^{[l-1]} f^{[l]2}\right] C_{\text{out}}^{[l]}(QR) +}_{\text{binary operations}} \\ & \underbrace{\sum_{l=1}^{L} N_{\text{h}}^{[l-1]} N_{\text{w}}^{[l-1]} \left[C_{\text{in}}^{[l-1]} f^{[l]2}\right] C_{\text{out}}^{[l]}(1-QR)}_{\text{floating point operations}}. \end{aligned} \tag{5.27}$$

Similarly, the PUM's complexity is determined by the cost of the feed-forward pass of the shallow CNN, as shown in Table 5.2 and the *'log'* barrier that form the barrier term.

$$C_{\text{PUM}} = \sum_{l=1}^{L_{\text{cv}}} N_{\text{h}}^{[l-1]} N_{\text{w}}^{[l-1]} \left[C_{\text{in}}^{[l-1]} f^{[l]2}\right] C_{\text{out}}^{[l]} + \sum_{j=1}^{L_{\text{fc}}} \left(2N_{\text{in}}^{[j-1]} + 1\right) N_{\text{out}}^{[i]} + C_{\text{log-barrier}} \tag{5.28}$$

where L_{cv} and L_{fc} are the number of convolution and fully connected layers, respectively. Based on the matrix/vector multiplications, the square absolute and l_2 norm values, the number of arithmetic operations involved in computing the terms in the *'log'* barrier functions for SLP-DNet and robust SLP-DNet are obtained as $4N_t^2K + 2N_tK + K$ and $8N_t^2K + 4N_tK + 6K$, respectively.

Table 5.3: Complexity analysis of proposed SLP-DSQNet and benchmark SLP schemes

Method	Arithmetic Operations (term; $\bar{n} = \mathscr{O}(2KN_t)$)	Complexity Order ($\bar{n} = N_t = K$)
Conventional BLP	$\sqrt{(4N_t+K+2)}\left[\bar{n}(2N_t+1)+\bar{n}(2N_t+1)^2+\bar{n}(K+1)^2+\bar{n}^3\right]\ln\left(\frac{1}{\varepsilon}\right)$	$\mathscr{O}(\bar{n}^{6.5})$
SLP Optimisation-based	$\sqrt{2N_t+1}\left[\bar{n}(2N_t+1)+\bar{n}(2N_t+1)^2+\bar{n}^3\right]\ln\left(\frac{1}{\varepsilon}\right)$	$\mathscr{O}(\bar{n}^{6.5})$
SLP-DNet	$2704K^2N_t+4N_t^2K+430KN_t-K$	$\mathscr{O}(\bar{n}^3)$
SLP-DBNet	$127K^2N_t+4N_t^2K+7KN_t-K-\frac{7}{8}$	$\mathscr{O}(\bar{n}^3)$
SLP-DTNet	$271K^2N_t+4N_t^2K+\frac{77}{2}KN_t-K-\frac{7}{8}$	$\mathscr{O}(\bar{n}^3)$
SLP-DSQBNet	$2704K^2N_t+430KN_t+4N_t^2K-K-\left[2577K^2N_t+423KN_t+\frac{7}{8}\right]\times QR$	$\mathscr{O}(\bar{n}^3)$
SLP-DSQTNet	$2704K^2N_t+430KN_t+4N_t^2K-K-\left[2433K^2N_t+\frac{783}{2}KN_t+\frac{7}{8}\right]\times QR$	$\mathscr{O}(\bar{n}^3)$
Robust Conventional BLP	$\sqrt{2K(2N_t+1)}\left[\bar{n}K(2N_t+1)^3+\bar{n}^2K(2N_t+1)^2+\bar{n}^3\right]\ln\left(\frac{1}{\varepsilon}\right)$	$\mathscr{O}(\bar{n}^{7.5})$
Robust SLP Optimisation-based	$\sqrt{2(2N_t+1)}\left[2\bar{n}K(2N_t+1)^2+\bar{n}^3\right]\ln\left(\frac{1}{\varepsilon}\right)$	$\mathscr{O}(\bar{n}^{6.5})$
Robust SLP-DNet	$2704K^2N_t+8N_t^2K+432KN_t+8N_t^2K+6K-2$	$\mathscr{O}(\bar{n}^3)$
Robust SLP-DBNet	$127K^2N_tK+8N_t^2K+9KN_t+6K-\frac{9}{8}$	$\mathscr{O}(\bar{n}^3)$
Robust SLP-DTNet	$271K^2N_t+8N_t^2K+\frac{81}{2}KN_t+6K-\frac{9}{8}$	$\mathscr{O}(\bar{n}^3)$
Robust SLP-DSQBNet	$2704K^2N_t+8N_t^2K+432KM+6K-2-\left[2577K^2N_t+423KN_t+\frac{7}{8}\right]\times QR$	$\mathscr{O}(\bar{n}^3)$
Robust SLP-DSQTNet	$2704K^2N_t+8N_t^2K+432KN_t+6K-2-\left[2433K^2N_t+\frac{783}{2}KN_t+\frac{7}{8}\right]\times QR$	$\mathscr{O}(\bar{n}^3)$

Finally, we use the information in Tables 5.2 and 5.1 along with (5.27) and (5.28) to obtain the complexity of SLP-DSQBNet as follows

$$C_{\mathrm{SQB}} = 2704K^2N_t + 430KN_t + 4N_t^2K - K - \left[2577K^2N_t + 423KN_t + \frac{7}{8}\right] \times QR. \tag{5.29}$$

We can obtain SLP-DSQTNet's complexity from (5.29) by introducing additional '0' state, and this additional bit yields

$$C_{\mathrm{SQT}} = 2704K^2N_t + 430KN_t + 4N_t^2K - K - \left[2433K^2N_t + \frac{783}{2}KN_t + \frac{7}{8}\right] \times QR. \tag{5.30}$$

We observe that by substituting $QR = 0$ in (5.29) or (5.30), we can obtain the complexity of SLP-DNet. Similarly, the complexities of SLP-DBNet and SLP-DTNet are also found by substituting $QR = 1$ in (5.29) and (5.30), respectively. Table 5.3 shows the complexities of the proposed and benchmarks precoding schemes. For illustration, we use the case of symmetry, where $(N_t = K = \bar{n})$, and show that our proposals have a considerably lower computational complexity of $\mathscr{O}(\bar{n}^3)$. In contrast, the optimisation-based SLP and conventional BLP methods have $\mathscr{O}(\bar{n}^{6.5})$ and $\mathscr{O}(\bar{n}^{7.5})$ computational complexities, respectively. While our proposed schemes have the same order of complexity as SLP-DNet (see Table 5.3), the number of arithmetic operations involved in their computations is lower than that of the SLP-DNet due to the presence of binary operations.

Table 5.4: Simulation settings

Parameters	Values
training Samples	50000
Batch Size (B)	200
Test Samples	2000
training SINR range	0.0dB - 45.0dB
Test SINR range (*i*-th user SINR)	0.0dB - 35.0dB
Optimiser	SGD with Adam
Initial Learning Rate, η	0.001
Learning Rate decay factor, α	0.65
Lower bit Activation	bits-width, $k = 2$
Number of blocks in the PUM	$B_l = 3$
training Iterations in the PUM per block	20
training iterations for the APM	10

5.7 Simulation Settings

We consider a downlink situation in which the BS is equipped with four antennas ($N_t = 4$) that serve K single users; and assume a single cell. We obtain the dataset from the channel realisations randomly generated from a normal distribution with zero mean and unit variance. The dataset is reshaped and converted to real number domain using the following expression $\Phi = \left[\hat{\mathbf{h}}_R \;\; \hat{\mathbf{h}}_I\right]^T$ as summarised in Figure 5.4. The input dataset is normalised by the transmit data symbol so that data entries are within the nominal range, potentially aiding the training. We generate 50,000 training samples and 2000 test samples, respectively. The transmit data symbols are modulated using a QPSK modulation scheme. The training SINR is obtained random from uniform distribution $\Gamma_{\text{train}} \sim \mathcal{U}(\Gamma_{\text{low}}, \Gamma_{\text{high}})$. Stochastic gradient descent is used with the Lagrangian function as a loss metric. A parametric rectified linear unit (**PReLu**) activation function is used for both convolutional and fully connected layers in a full-precision SLP-DNet and the low-bit activation function (5.19) for SLP-SQDNet. After every iteration, the learning rate is reduced by a factor $\alpha = 0.65$ to help the learning algorithm converge faster. The models are implemented in Pytorch 1.7.1 and Python 3.7.8 on a computer with the following specifications: Intel(R) Core (TM) i7-6700 CPU Core, 32.0GB of RAM. Tables 5.4 summarises the simulation parameters depict the NN component settings of the

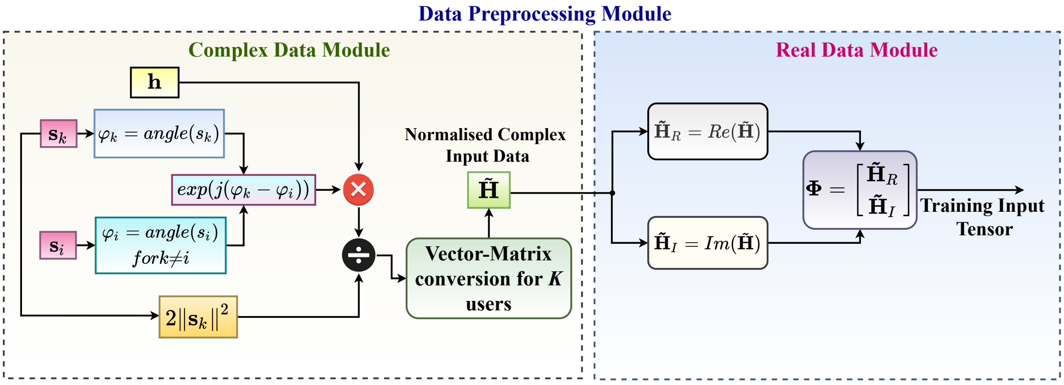

Data Preprocessing Module
Complex Data Module
Real Data Module
$\mathbf{h}$
$\mathbf{s}_k$
$\varphi_k = angle(s_k)$
$exp(j(\varphi_k - \varphi_i))$
$\mathbf{s}_i$
$\varphi_i = angle(s_i)$
$for k \neq i$
$2\|\mathbf{s}_k\|^2$
Normalised Complex Input Data
$\tilde{\mathbf{H}}$
Vector-Matrix conversion for K users
$\tilde{\mathbf{H}}_R = Re(\tilde{\mathbf{H}})$
$\tilde{\mathbf{H}}_I = Im(\tilde{\mathbf{H}})$
$\mathbf{\Phi} = \begin{bmatrix}\tilde{\mathbf{H}}_R \\ \tilde{\mathbf{H}}_I\end{bmatrix}$
Training Input Tensor

Figure 5.4: Schematic diagram of dataset generation and preprocessing

SLP-DNet of the robust SLP-DNet.

5.8 Simulation Results and Discussion

In the following set of results we compare our proposed quantised DL-based SLP scheme's performance against its corresponding full-precision (SLP-DNet) counterpart's (see Chapter 4) and other benchmark schemes, such as conventional BLP [73, 77] and the SLP optimisation-based [49]. Primarily, we design full low-bit binary and ternary SLP-DNet models (SLP-DBNet and SLP-DTNet), where the real-valued weights and activation are constrained to 1-bit. Similarly, the expressive learning abilities of SLP-DBNet and SLP-DTNet are further enhanced by designing their corresponding low-bit hybrid stochastically quantised versions (SLP-DSQBNet and SLP-DSQTNet), where part of the weight matrix is quantised to a lower bit, while the remaining is left in its 32-bit floating-point precision. The resulting weight matrix is a hybrid containing both binary and real-valued entries with the activations all reduced to 2-bit according to (5.19).

5.8.1 Performance Evaluation of QSLP-DNet and SLP-DNet

The performances of SLP-DBNet, SLP-DTNet, SLP-DSQBNet, SLP-DSQTNet for $QR = 0.5$ against SLP-DNet and other benchmark precoding schemes (conventional BLP, SLP optimisation-based) are shown in Figure 5.5. It can be observed that both SLP-DBNet and SLP-DTNet have higher transmit power than the SLP optimisation-based and SLP-DNet schemes. Therefore, SLP optimisation-based

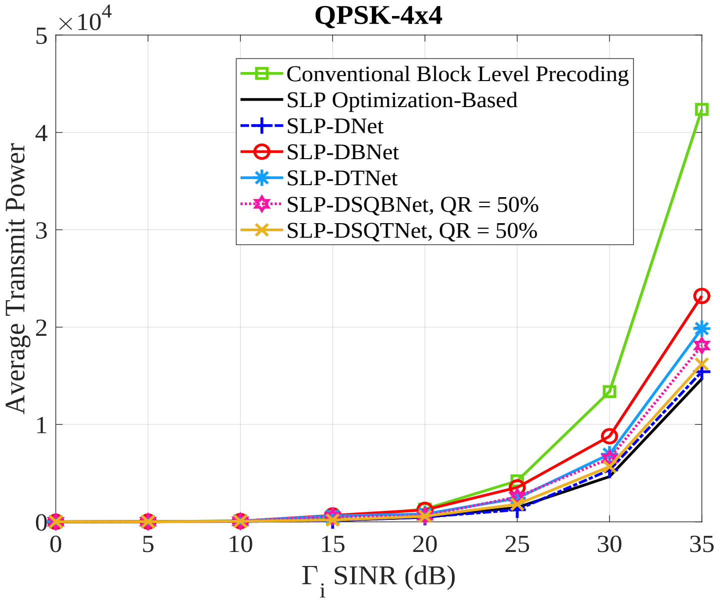


Figure 5.5: Transmit Power vs SINR averaged over 2000 test samples for Conventional Block Level Precoding, SLP optimisation-based and nonrobust quantised learning-based SLP solutions, $N_t = 4$, $K = 4$ and $QR = 50\%$

and SLP-DNet solutions require less power to transmit the same amount of data symbols than SLP-DBNet and SLP-DTNet. The loss in performance is expected because some information is lost during feed-forward weight/input convolutions due to quantisation and the inhomogeneous nature of the quantisation errors.

Furthermore, a closer examination of Figure 5.5 reveals that the SLP-DSQBNet and SLP-DSQTNet offer less transmit power than their corresponding full binary and ternary versions. Our simulation also shows that learning by stochastic quantisation results in the performance close to the full-precision learning model (SLP-DNet) with a significant model size reduction (memory savings at the inference), as we shall see later. We argue that the decrease in the available transmit power at the BS in this scenario is because not all the weights matrix rows are quantised at once. The quantisation error is used to direct the gradient descent towards the best local minima during training. Accordingly, we find that at 30dB SINR, the performance of SLP-DBNet and SLP-DTNet falls by 58% and 35% of the SLP optimisation-based solution, respectively. On the other hand, the performance gaps of SLP-DSQBNet, SLP-DSQTNet, and SLP-DNet are 22.2%, 9.62%, and 5% of the SLP optimisation-based solution, respectively. Therefore, while

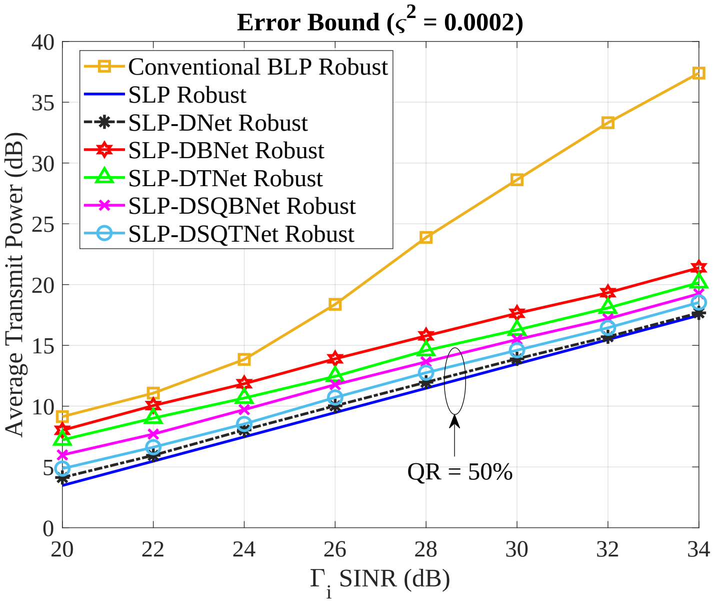


Figure 5.6: Transmit Power vs SINR averaged over 2000 test samples for conventional, SLP optimisation-based and robust quantised learning-based SLP solutions under $N_t = 4$, $K = 4$ and $\varsigma^2 = 0.0002$

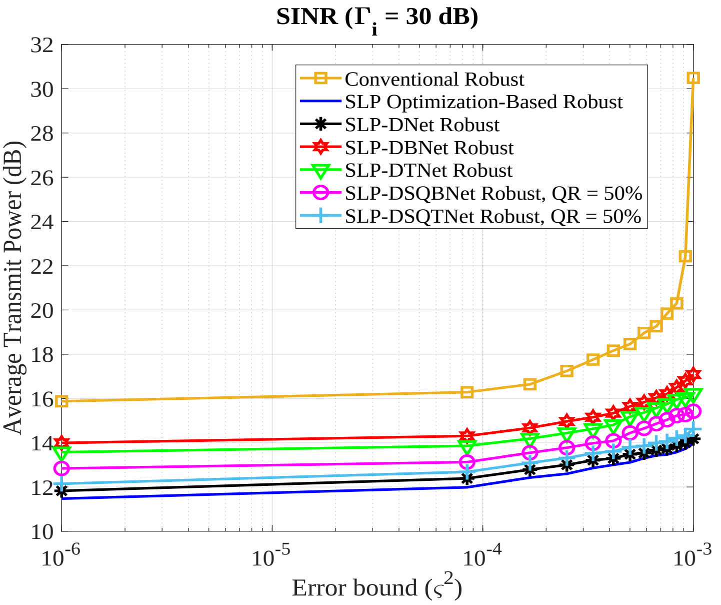


Figure 5.7: Transmit Power vs Error-bound for Conventional BLP, robust SLP optimisation-based and robust quantised learning-based SLP solutions under $N_t = 4$, $K = 4$ and $QR = 50\%$

the fully quantised model's accuracy is significantly low, the stochastically hybrid quantised counterparts and full-precision models' accuracy is within $88\% - 96\%$ of the optimal solution.

5.8.2 Performance of Robust SLP-SQDNet

Figures 5.6 and 5.7 compare the performances of SLP-SQDNet and the traditional CSI-robust precoder for the 4×4 MISO system. Figure 5.6 depicts how the average transmit power increases with the *SNR* thresholds, for CSI error bounds $\varsigma^2 = 10^{-4}$ and $QR = 50\%$. The robust SLP optimisation-based and SLP-DNet are observed to show a significant power savings of about 60% and 58%, respectively compared to the robust conventional BLP. Similarly, proposed fully quantised learning-based precoders (SLP-DBNet and SLP-DTNet) portray similar transmit power reduction trend. They show considerable power savings of $40\% - 58\%$ against the conventional optimisation result. While the fully quantised models have demonstrated substantial performance loss compared to SLP-based optimal precoder, SLP-DSQBNet and SLP-DSQTNet offer $90\% - 98\%$ striking performance correlation with the SLP optimisation-based optimal solutions, respectively.

Furthermore, we investigate the effect of the CSI error bounds on the transmit power at 30dB SINR. Figure 5.7 depicts the variation of the transmit power with increasing CSI error bounds. Moreover, a significant increase in transmit power can be observed where the channel uncertainty lies within the region of CSI error bounds of $\varsigma^2 = 10^{-3}$. Interestingly, like the SLP optimisation-based algorithm, by exploiting the CI, the proposed unsupervised learning methods also show a descent or moderate increase in transmit power. To further understand the impact of the *QR* on the transmit power, Figure 5.8 compares the performance of the proposed stochastic quantisation learning-based CI-nonrobust precoders evaluated at 30dB. In this scenario, we observe that the average transmit power available at the BS required to transmit data symbols increases as more weights and activations are quantised. This is true because the network performance accuracy gradually improves as more weights with full floating-point values are introduced.

5.8.3 Complexity and Memory Evaluation

The proposed learning schemes' complexities are examined in two folds: firstly, we compare the number of FLOPs operations involved in our proposed learning methods and those of the benchmark precoding schemes'. Secondly, we evaluate and

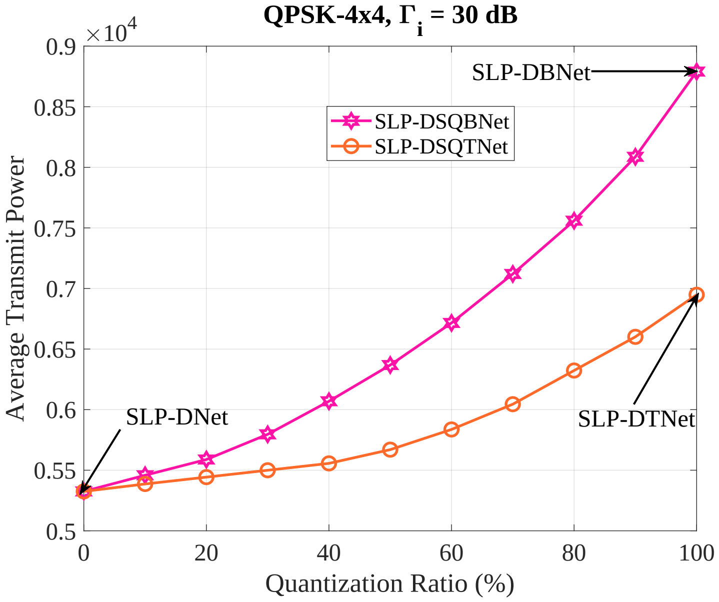


Figure 5.8: Transmit Power vs Quantisation ratio averaged over 2000 test samples for non-robust SQ SLP-DNet models and full-precision SLP-DNet model under $N_t = 4$, $K = 4$ and $\Gamma = 30$ dB

assess the inference memory requirements of our proposed learning-based precoding techniques.

5.8.3.1 Number of FLOPs Operations

The computational costs of the SLP-DNet are obtained from the PUM and the feed-forward convolutions of the CNN that makes up an APM. For the PUM, the dominant computational cost comes from computing the proximal barrier term (Chapter 4). It can be seen that both SLP optimisation-based algorithm and the proposed learning schemes are feasible for all sets of N_t BS antennas and K mobile users. However, for conventional BLP, the solution is only feasible for $N_t \geq K$.

Figure 5.9 (a) shows the number of FLOPs operations of the proposed unsupervised learning solutions per symbol for nonrobust formulations. The dominant operations involved in SLP-DNet at the inference are matrix-matrix or vector-matrix convolution. The gap in the computational cost between SLP-DNet and SLP optimisation-based methods increases with the growing number of mobile users. For example, we find that the complexity of SLP-DNet is $\sim 10\times$ lower than SLP optimisation-based at $K = 10$, while that of SLP-DSQBNet and SLP-DSQTNet

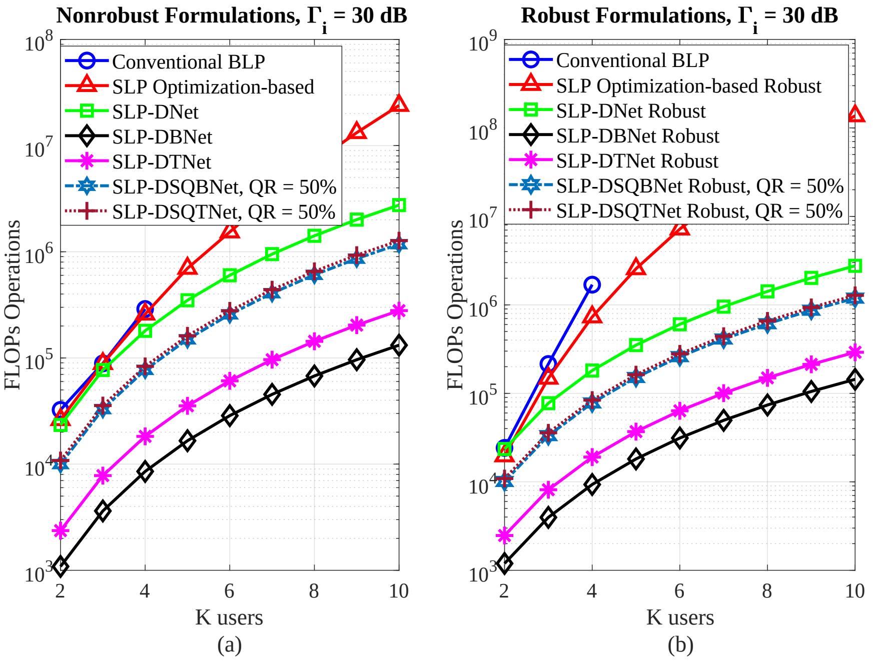
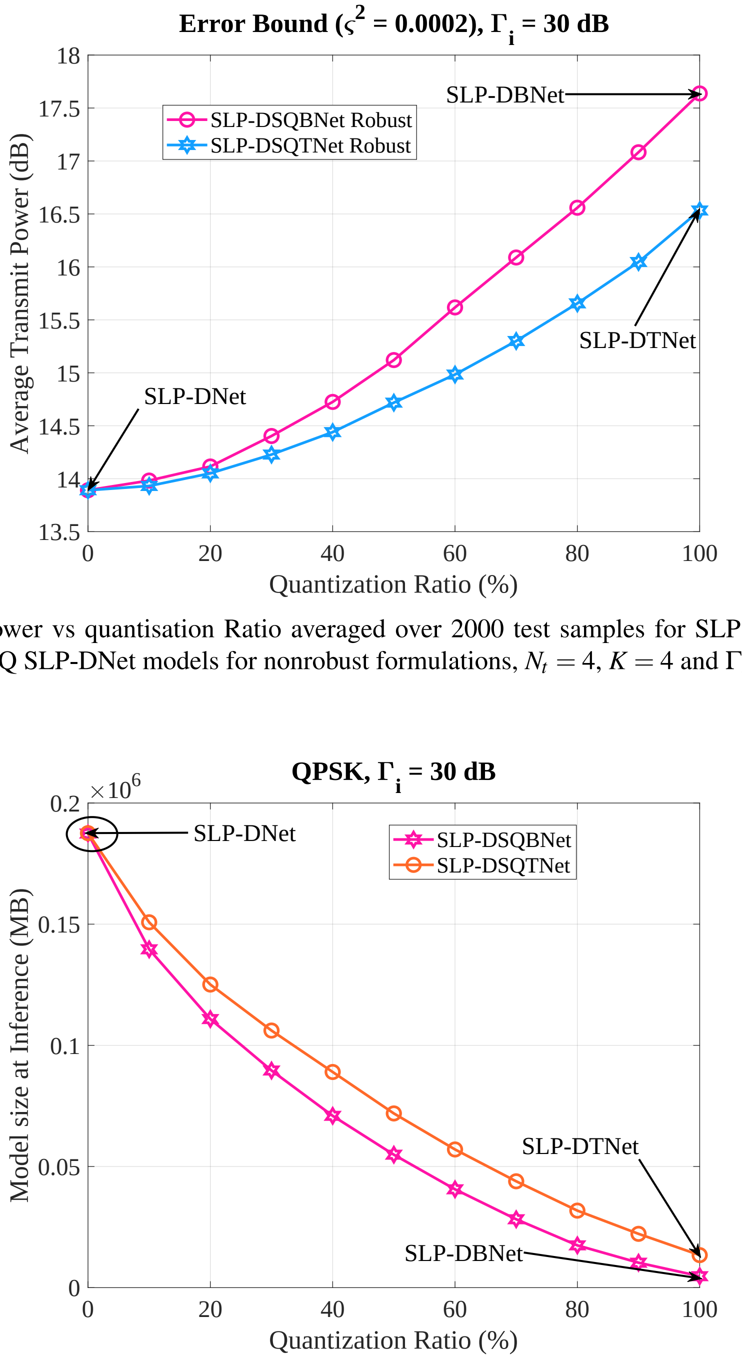


Figure 5.9: Comparison of FLOPs operations performed for Nonrobust and Robust precoding schemes, i.e, conventional BLP, SLP optimisation-based and SLP learning-based models using four BS antennas ($N_t = 4$) and $QR = 50\%$

are $\sim 20\times$ much lower due to the presence of binary operations. Furthermore, SLP-DBNet and SLP-DTNet offer an additional computational complexity reduction than SLP-DSQBNet and SLP-DSQTNet because binary bit-wise operations replace the entire MACs calculations in the forward pass. It is important to recall that SLP-DTNet outperforms SLP-DBNet in all scenarios. However, we observe that SLP-DTNet is slightly slower than SLP-DBNet, and this is due to the additional '0' binary state introduced in the former. We also note that the advantages of the SLP-DBNet and SLP-DTNet are further enhanced via stochastic quantisation but at the expense of small additional complexity overhead. The same trend is also observed in the case of a robust channel scenario, as shown in Figure 5.9 (b).

Accordingly, we can deduce that while fully binarised DNN could offer significant training and inference accelerations, it could otherwise lead to significant performance degradation. However, quantising the weight matrix via a stochastic channel selection based on the quantisation error leads to improved performance in terms of reduction in transmission power. Therefore, we can conclude that the results in Figures 5.9(a) and 5.9 (b) demonstrate that the proposed quantised DL-based

SLP solutions offer a good trade-off between the performance and computational complexity.



(a) Transmit Power vs quantisation Ratio averaged over 2000 test samples for SLP optimisation-based and SQ SLP-DNet models for nonrobust formulations, $N_t = 4$, $K = 4$ and $\Gamma_i = 30$ dB

(b) Memory requirement at Inference vs quantisation ratio for SLP-based Full-precision SLP-DNet, SLP-DSQBNet and SLP-DSQTNet under $N_t = 4$, $K = 4$ and $\Gamma_i = 30$ dB

Figure 5.10: Average power and inference memory requirement vs quantisation error of the proposed learning-based precoding schemes

5.8.3.2 Model Size and Memory Utilisation

Generally, GPU can speedup the offline training of DNNs. However, most modern GPUs are memory-constrained (e.g.GTX 980: 4GB, Tesla K40: 12GB, Tesla K20: 5GB and GTX Titan X: 12GB)[178]. Practically, the size of the DNN is often bounded by the available memory. Therefore, it is beneficial to estimate the memory requirements of the DNN at the inference. Likewise, the actual memory utilisation also depends on the implementation. Here, we examine and analyse the memory utilisation of full-precision SLP-DNet and its corresponding quantised versions at inference. By memory utilisation, we refer to the model size at the testing phase. For this analysis, we adopt the approach presented in [179] to calculate the inference memory utilisation as the summation of 32-bit times the number of floating-point parameters and 1-bit times the number of binary parameters. Mathematically, this can be expressed as $\frac{1}{32}W_b + W_f$, where W_b and W_f are the binary and floating-point weights, respectively.

Figure 5.10a shows the average transmit power vs quantisation ratio (i.e. the proportion of weights that are quantised) at 30dB SINR. The average power at $QR = 0$ corresponds to SLP-DNet while $QR = 1$ represents the corresponding fully quantised counterparts (SLP-DBNet and SLP-DTNet). Moreover, the transmit power gradually increases as more weights are quantised. It is important to note that for a unit quantisation ratio ($QR = 1.0$), all the weights are 100% quantisation, where the model could be either a typical binary or ternary. On this note, it is clear that the SLP-DSQTNet offers less transmit power than SLP-SQDBNet. We find that quantising half of the weights ($QR = 50\%$) could guarantee a good performance within $80\% - 98\%$ of the full-precision model for both SLP-SQDBNet and SLP-DSQTNet, respectively. To investigate the amount of the memory required at inference with the increase in the quantisation ratio, we plot the model size vs QR as depicted in Figure 5.10b. We find that less memory is required as the quantisation moves towards extreme binarization to the right of the QR-axis. It can be seen that the continuous line represents a full-precision SLP-DNet (i.e., $QR = 0$), while $QR = 1$ represents a fully quantised model.

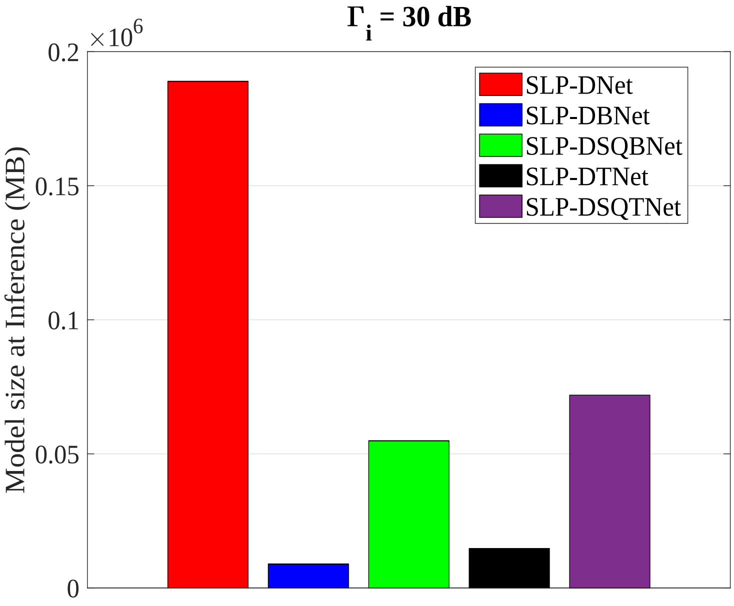


Figure 5.11: Memory requirement at Inference for Full-precision SLP-DNet, SLP-DBNet, SLP-DTNet, SLP-DSQBNet and SLP-DSQTNet under $N_t = 4$, $K = 4$, $\Gamma_i =$ 30dB and $QR = 50\%$

Table 5.5: Inference memory utilisation

Models	Weights	Activations	Memory usage (MB)	Memory saving
SLP-DNet	$(32-\text{bit}) \in \mathbb{R}$	$(32-\text{bit}) \in \mathbb{R}$	0.1898	–
SLP-DBNet	$\{-1,+1\}$	$\{-1,+1\}$	0.0089	21.33×
SLP-DTNet	$\{-1,0,+1\}$	$\{-1,+1\}$	0.0146	13×
SLP-DSQBNet	$\{-\beta_{\text{qf}},\beta_{\text{qf}}\}$	$\{-\beta_{2-bit},\beta_{2-bit}\}$	0.0548	3.46×
SLP-DSQTNet	$\{-\beta_{\text{qf}},0,\beta_{\text{qf}}\}$	$\{-\beta_{2-bit},\beta_{2-bit}\}$	0.0719	2.64×

Furthermore, Figure 5.11 shows that SLP-DBNet and SLP-DBNet provide considerable memory savings up to $\sim 21\times$ and $\sim 13\times$ compared to the full-precision SLP-DNet because the extreme quantisation reduces the available learning parameters significantly. This brings about a trade-off between performance and model size, which is compensated by hybrid quantisation as in SLP-DSQBNet and SLP-DSQTNet. Table 5.5 presents the summary of the inference memory requirements, MACs, and binary operations of different proposed learning implementations. For SLP-DSQBNet and SLP-DSQTNet, the weights are constrained to the following quantisation $\{-\beta_{\text{qf}},\beta_{\text{qf}}\}$ and $\{-\beta_{\text{qf}},0,\beta_{\text{qf}}\}$ while the activations are clipped to $\{-\beta_{2-bit},\beta_{2-bit}\}$ $2-$bit quantised values, respectively. This shows that the hybrid quantisation enhances the representational capabilities of the convolutional blocks.

5.9 Summary

In this chapter, we investigate binary and ternary weight quantisation techniques for DNN model compression. For typical binary and ternary quantisations, the real-valued NN weights are converted to binary and ternary values (i.e. -1, 0 and 1), allowing the operations between the inputs and weights tensors to be performed in binary format. We propose a hybrid quantisation DNN-based SLP scheme termed (SLP-QSDNet) based on binary and ternary operations for power minimisation for a multi-user downlink MISO system. The proposed quantised precoding schemes are extensions of the model-driven unsupervised learning frameworks derived from the proximal IPM barrier *'log* function for a relaxed phase rotation described in Chapter 4. We showed that the proposed approach resulted in fast online learning and a significant model size reduction, which could help render the trained model memory-efficient during deployment on the device's edge. Overall, our proposed approaches provide a scalable tradeoff between performance and complexity in learning-based SLP schemes for a MU-MISO downlink transmission.

Chapter 6

Complexity-Scalable Neural Network Based MIMO Detection With Learnable Weight Scaling

6.1 Introduction

The chapter mainly focuses on signal processing at the receiver side. Signal separation (detection) is one of the principal implementation difficulties of the MIMO technology at the receiver side due to the co-channel interference. Several ingenious techniques with a viable computational complexity, including ML-based methods, have been proposed to subdue this difficulty. In this, we introduce a scalable, low complexity DNN design MIMO detection scheme. MIMO detectors have been extensively studied over the last two decades with the view to improving their detection accuracy and decreasing their complexities [6, 17, 19, 180–182].

6.1.1 Related Works

The state-of-the-art learning based iterative detectors, such as OAMP-Net and TPG-Net described in Chapter 2.6.5 channel inversion in every training iteration. To address this drawback, we propose a generic weight-scaling NN (WeSNet) framework for reducing the complexity of broader DNN-based receivers and therefore extends to numerous relevant NN designs that do not embed NN architecture.

Generally, designing deeper NN architectures for signal detection problems comes with significantly increased training and inference complexity, while gains in detection performance are not always significantly increased. This creates the imperative for systematic approaches to design DNN architectures with scalable complexity that can speed up offline training (learning),[1] facilitate model deployment and inference on a range of devices such as mobile devices, and other embedded hardware platforms with limited resources. Popular techniques of complexity reduction that are similar to our proposal in style are Dropout [183], Drop-Connect [184] and Pruning [171]. However, these schemes fundamentally differ from our proposal because most of them are used to prevent overfitting, and they are not explicitly designed (of applicable) for complexity reduction. The simplest of them is Dropout [2] where some units (neurons) are randomly shot during training. At inference time however, Dropout uses the full network whereas our proposed framework allows for the network to dynamically adjust its computational complexity and detection accuracy characteristics at inference. While many proposals have been put forward for accelerated DNN training and inference in computer vision [170, 177, 184], to the best of our knowledge, no systematic DNN acceleration has so far been designed for physical layer communications. In this work, we attempt to fill this gap by proposing a complexity-scalable DNN model for efficient MIMO detection.

In this thesis, we introduce the concept of monotonic non-increasing profile function that scale each layer of the NN in order to allow the network to dynamically learn the best attenuation strategy for its own weights during training. By doing so, we introduce sparsity in the DNN, which results in a significant complexity saving at inference. Our focus is on DNN designs that unfold iterative projected gradient descent unconstrained optimisation for massive MIMO ML-based detection. While

[1] In practice, training a DNN is done offline and is computationally expensive in addition to requiring large training data. Generally, the performance of a trained DNN model is determined by its ability to generalise well on a new set of data (test data). Therefore, model testing is done online using Monte Carlo simulation with new channel instantiations at different SNR conditions at the edge of the device to evaluate the efficacy of the trained model.

[2] Dropout requires additional matrices for dropout masks, random selection of numbers for each entry of these matrices and matrix multiplication of the masks with the corresponding weights. At inference time, which is the focus of our work, Dropout uses the full network and does not allow for scalable complexity-accuracy adjustments.

methods of artificial “suppression” of neurons during training are known to create sparsity and can be detrimental to inference accuracy [171, 185], we show that, by tuning these profile function appropriately, we can provide a control mechanism that trades off DNN complexity for detection accuracy in a scalable manner. Our contributions are summarised below:

- We introduce a weight scaling framework for DNN-based MIMO detection. Our approach is realised by adjusting layer weights through monotonic profile functions. The original DNN design is based on DetNet, i.e., unfolding a projected gradient descent scheme [22]. We term our proposal the weight-scaling neural-network based MIMO detector (WeSNet).

- In order to allow for entire layers to be abrogated in a controllable manner during inference, we introduce a regularisation approach that imposes constraints on the layer weights. This allows for scalable reduction in the model size and the incurred computational complexity, with graceful degradation in the detection accuracy.

- To improve the performance of WeSNet, we introduce a learnable accuracy-complexity design, where the weight profile functions themselves are made trainable in order to prevent vanishing gradients due to changes in the values of activations. This improves the detection accuracy of the WeSNet at the cost of increased memory due to increase in the model parameters.

- Finally, we present a comprehensive complexity analysis of WeSNet inference in relation to learning-based MIMO detector(DetNet) and traditional detectors. Our study and results show that under the same experimental conditions, WeSNet with 50% of the layer weights outperforms the detection accuracy of DetNet while offering 51.43% reduction in complexity and close to 50% reduction in model size. Furthermore, its detection accuracy is similar to SDR with nearly 10-fold reduction of computational complexity.

6.2 System Model

Consider a communications system with N_t transmit and N_r receive antennas. The received signal is modelled using a standard MIMO channels equation (2.1) described in Chapter 2. For convenience and ease of implementation, we will use the equivalent MIMO model in real domain as defined in Chapter 2.6.3. The transmitted symbols can be recovered by minimising the Euclidean distance between the received and the transmitted symbols

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\mathbf{S}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2^2 \tag{6.1}$$

where $\mathbf{S}$ is the constellation set defined by the modulation scheme used (BPSK, 4-QAM and 16-QAM).

The premise of the operation of all learning-based detectors is that the estimate of the received symbols is obtained from a trained network by an update rule using an iterative projected gradient descent formulation [117]. For a function defined by $f(x,y)$, the estimate of $\mathbf{x}$ and $\mathbf{y}$ over the r-th iteration (i.e., layer) can be found from gradient descent using the following update rule:

$$\mathbf{x}_{r+1} = \mathbf{x}_r - \eta \frac{\partial f(x,y)}{\partial x} \tag{6.2a}$$

$$\mathbf{y}_{r+1} = \mathbf{y}_r - \eta \frac{\partial f(x,y)}{\partial y} \tag{6.2b}$$

where η is the learning rate.

DetNet is designed by applying gradient descent optimisation in (6.1) expressed as

$$\hat{\mathbf{s}}_{r+1} = \hat{\mathbf{s}}_r - \eta_r \frac{\partial \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2^2}{\partial \mathbf{s}}\bigg|_{\mathbf{s}=\hat{\mathbf{s}}_r} \tag{6.3}$$

Simplifying (6.3), we obtain:

$$\hat{\mathbf{s}}_{r+1} = \hat{\mathbf{s}}_r - 2\eta_r \mathbf{H}^T\mathbf{y} + 2\eta_r \mathbf{H}^T\mathbf{H}\hat{\mathbf{s}}_r \tag{6.4}$$

By using $\mathbf{H}^T\mathbf{y}$, $\mathbf{H}^T\mathbf{H}$ and $\mathbf{s}$ as inputs, and via the application of non-linear functions prior to the outputs, the formulation of (6.4) is converted to three sublayers with each sublayer comprising a perceptron, also known as fully-connected NN. This is defined by the following equations

$$\mathbf{u}_r = \Theta(\mathbf{W}_{1r}\mathbf{x}_r + \mathbf{b}_{1r}) \tag{6.5}$$

where:

$$\mathbf{x}_r = \Sigma(\mathbf{H}^T\mathbf{y},\ \mathbf{H}^T\mathbf{H}\mathbf{s}_r,\ \mathbf{s}_r,\ \mathbf{a}_r) \tag{6.6}$$

$$\hat{\mathbf{s}}_{r+1} = \psi(\mathbf{W}_{2r}\mathbf{u}_r + \mathbf{b}_{2r}) \tag{6.7}$$

$$\hat{\mathbf{a}}_{r+1} = \mathbf{W}_{3r}\mathbf{u}_r + \mathbf{b}_{3r} \tag{6.8}$$

$\Sigma(\cdot)$ is the concatenation function, and $\Theta(\cdot)$ and $\psi(\cdot)$ are nonlinear and piece-wise linear sign functions, respectively, and subscripts $1r$, $2r$ and $3r$ indicate the three sublayers of layer r. The trainable parameters that are optimised during training are defined by

$$\Psi = \{\mathbf{W}_{1r},\ \mathbf{W}_{2r}\ ,\ \mathbf{W}_{3r},\ \mathbf{b}_{1r},\ \mathbf{b}_{2r},\ \mathbf{b}_{3r}\}_{r=1}^{L} \tag{6.9}$$

6.3 Proposed Weight-Scaling NN based MIMO Detector (WeSNet)

In this section, we propose a scalable accuracy-complexity framework for DNN-based MIMO receivers through systematic weight scaling with monotonic non-increasing functions for both feed-forward and edge inference computations. This allows for our proposal to have minimum deployment friction as it allows for the best operating point in the accuracy-complexity sense to be devised at inference.

6.3.1 Weight Scaling Vector Coefficient (WSVC)

A WSVC is computed by applying monotonically non-increasing coefficients (known as profile function coefficients) to the layer weights during the forward

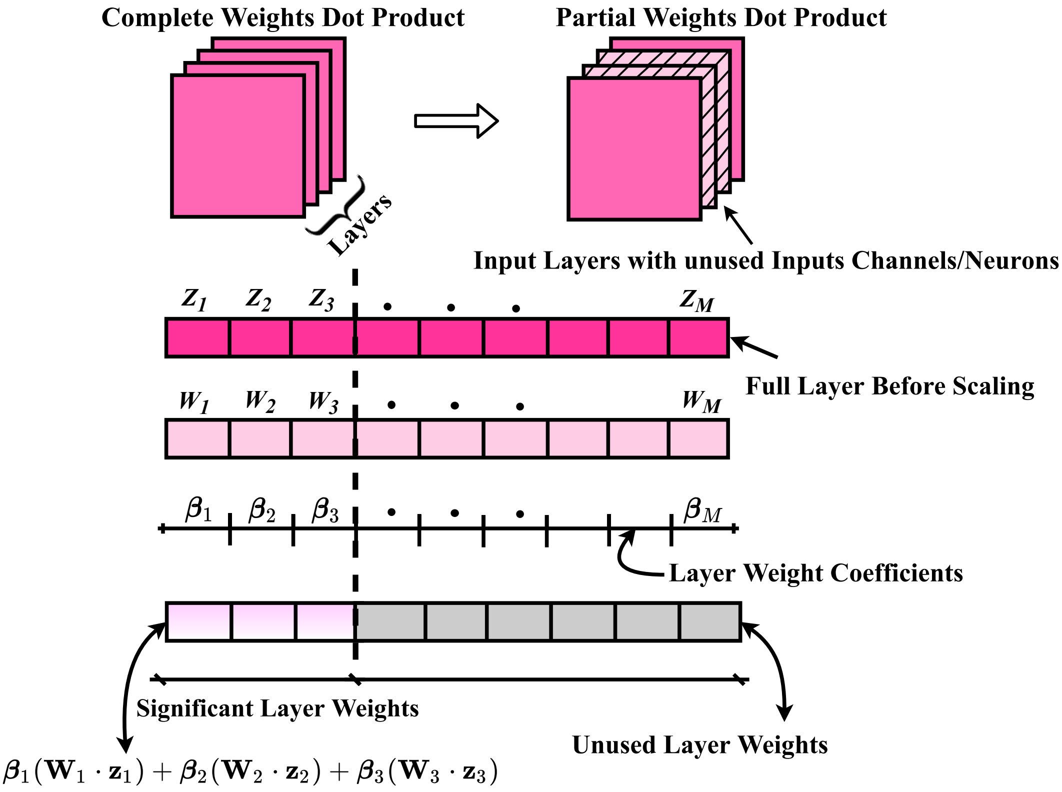

Complete Weights Dot Product
Partial Weights Dot Product
Layers
Input Layers with unused Inputs Channels/Neurons
Z_1 Z_2 Z_3 • • • Z_M
Full Layer Before Scaling
W_1 W_2 W_3 • • • W_M
β_1 β_2 β_3 • • • β_M
Layer Weight Coefficients
Significant Layer Weights
Unused Layer Weights
$\beta_1(\mathbf{W}_1 \cdot \mathbf{z}_1) + \beta_2(\mathbf{W}_2 \cdot \mathbf{z}_2) + \beta_3(\mathbf{W}_3 \cdot \mathbf{z}_3)$

Figure 6.1: WSVC in a single layer of an MLP allowing for attenuated layer weights to (optionally) be dropped.

propagation. This results in prioritising the selection of the layer weights in decreasing fashion from the most significant to least significant. Mathematically, for two given vectors, $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ and $\mathbf{y} = [y_1, y_2, \ldots, y_N]^T$, if β is the vector of the profile coefficients, WSVC is the pruned version of the form

$$\sum_{i=1}^{N} \beta_i x_i y_i = \beta_1 x_1 y_1 + \beta_2 x_2 y_2, \ldots + \beta_N x_N y_N \tag{6.10}$$

In a standard fully connected NN, the output of the feed forward pass is given by

$$z_j = \sum_{i=1}^{N} W_{ji} x_i + b_j \tag{6.11}$$

where i and j are the input and output dimensions (size of the neurons) respectively; x_i is the i-th input components, W_{ji} is the channel or layer weight corresponding to the jth output and b_j is the output bias. The corresponding WSVC is derived by

$$z_j = \sum_{i=1}^{N} \beta_i W_{ji} x_i + b_j \tag{6.12}$$

Figure 6.1 shows the difference between the feed forward computations of a layer of an MLP and the MLP augmented by WSVC. The part of the WSVC corresponding to significant layer weights is indicated by the light coloured shaded region on the bottom-right side of the figure. The example shows that, via the WSVC, we can compute and use only one-third of the channel/layer weights out of the N layer dimension, as the remaining two-thirds of the weights are attenuated and can be dropped.

6.4 Weight Coefficient Profile Function

We begin by introducing two non-increasing monotonic profile functions (Linear and Half-Exponential functions) for the weight coefficients [185] as shown in Figure 6.2.

6.4.1 Linear Profile Function

This function comprises the profile coefficients obtained from the linear equation of the form:

$$\beta_i = 1 - \frac{i}{N} \; ; \; \forall \, i = 1, \, 2, \, \ldots \, , \, N \tag{6.13}$$

where N is the layer size.

6.4.2 Half-Exponential Profile Function

This is a hybrid profile function from uniform and exponential functions. This function attenuates coefficients corresponding to half of the channel via an exponential decay function. The implication of this is that it allows the network training to adjust the gradient flow such that important weights are retained in the non-attenuated

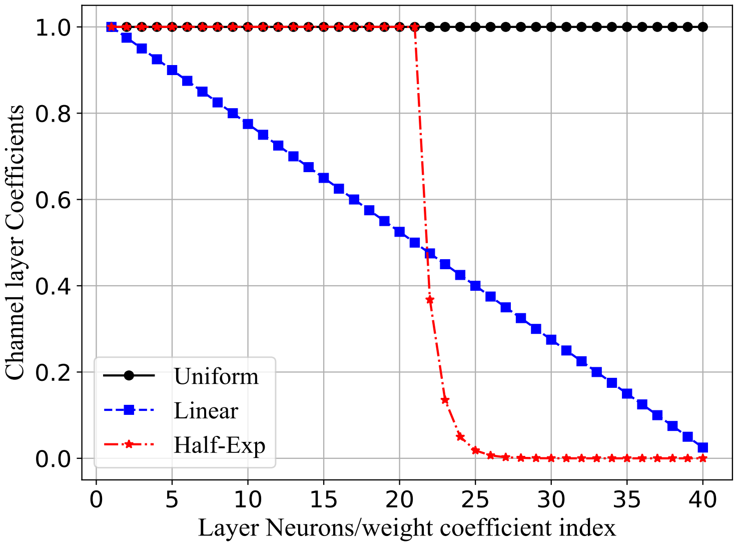


Figure 6.2: Profile coefficients vs Neurons/Layer weight index.

half of each layer and the less important ones in the exponentially-attenuated half.

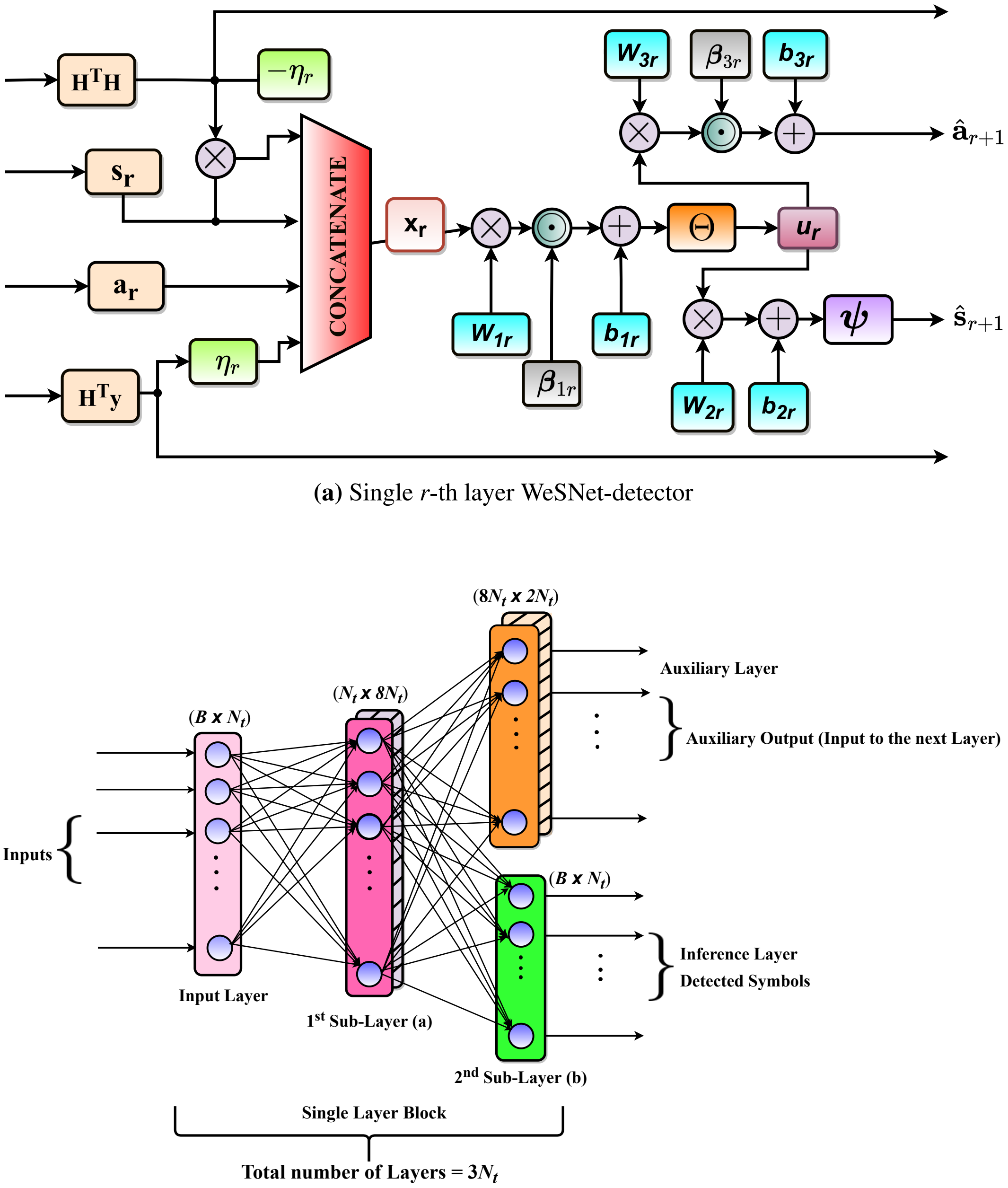
$$\beta_i = \begin{cases} 1 & if\ i\ \leq \frac{N}{2}\ \forall\ i\ =\ 1,\ 2,\ \ldots,N \\ exp\left(\frac{N}{2} - i - 1\right) & otherwise \end{cases} \tag{6.14}$$

6.4.3 Structure of the WeSNet-Detector

WeSNet is a nonlinear estimator designed by unfolding the ML metric using a recursive formulation of the projected gradient descent optimisation. Our proposed detector applies the profile coefficients on the existing DetNet. Such a modification reduces the computational complexity for training the detector. We apply profile coefficients to (6.5) and (6.8) to obtain the following non-linear WSVCs over the i-th and j-th inputs of the first and third sublayers of the r-th layer, respectively.

$$u_{j[r]} = \Theta\left\{\sum_{i=1}^{N} \beta_{i[1r]} W_{ji[1r]} x_{i[r]} + b_{j[1r]}\right\} \tag{6.15}$$

$$\hat{a}_{k[r+1]} = \sum_{j=1}^{M} \beta_{j[3r]} W_{kj[3r]} u_{j[r]} + b_{k[3r]} \tag{6.16}$$



(a) Single r-th layer WeSNet-detector

(b) Single Layer WeSNet Architecture

Figure 6.3: WeSNet Model Architecture

where: j and k are the outputs of the first and third sublayers of layers r and $r+1$ respectively, N and M are their corresponding sizes, and bracketed subscripts are added to explicitly indicate the membership of components to their corresponding network layers and sublayers.

WeSNet has $3N_t$ layers with each layer having three sub-layers, the input layer, the auxiliary and the detection layer. The layer weights of the last sub-layer (detection layer) described by (6.7) are not scaled in order to maintain the full dimension of the detected symbols as originally transmitted. The flowchart of a single-layer WeSNet based on the (6.7), (6.15) and (6.16) is shown in Figure 6.3a. The complete architecture of the WeSNet is shown in Figure 6.3b. Since the error estimation of the ML-detector does not require the knowledge of the noise variance, the loss function of WeSNet is derived as the weighted sum of the detector's errors normalised with the loss function of the standard linear inverse detector (ZF) as

$$\mathscr{L}(\mathbf{s};\hat{\mathbf{s}}(\mathbf{H},\mathbf{y}:\Psi)) = \sum_{r=1}^{L} \log(r) \frac{\|\mathbf{s}-\hat{\mathbf{s}}_r\|_2^2}{\|\mathbf{s}-\tilde{\mathbf{s}}\|_2^2} \tag{6.17}$$

6.5 Introducing Robustness through Regularised WeSNet (R-WeSNet)

In this section, we introduce log-regularisation with a sparsity-enforcing mechanism. Unlike other proposals that employ such mechanisms as the means to avoiding over-fitting, the combination of our log-regularisation with the proposed profile functions enables the network to learn the best profile function scaling to gracefully trade-off accuracy and complexity. Importantly, this achieves *scalable* accuracy-complexity operation at inference by simply discarding parts of network layers (or even entire layers).

6.5.1 Rationale

Given that our aim is to introduce sparsity in conjunction with our profile function coefficients so that layers (and parts of layers) with few non-zero coefficients can be removed to scale complexity, we propose the use of a *log*-l_1-norm. The choice of l_1-norm is motivated by the fact that it forces some of the coefficients to be zero and leads to sparsity [186], thereby making it more appealing and robust than l_2-norm, as well as a better candidate for feature selection.

6.5.2 Proposed Loss Function

Following the above motivation, the loss function of (6.17) is modified such that a *log*-l_1-norm penalty term is imposed on the weights:

$$\mathscr{L}(\mathbf{s};\hat{\mathbf{s}}(\mathbf{H},\mathbf{y}:\Psi)) = \sum_{r=1}^{L} \log(r) \frac{\|\mathbf{s}-\hat{\mathbf{s}}_r\|_2^2}{\|\mathbf{s}-\tilde{\mathbf{s}}\|_2^2} + \tilde{\lambda} f(\beta_r, \tilde{W}_r) \tag{6.18}$$

where $\tilde{\lambda}$ is the regularisation parameter that controls the importance of sparsity in the layers weights and $f(\beta_r, \tilde{W}_r)$ is the function of layer weights with respect to the neuron connections between adjacent layers, and is given by

$$f(\beta_r, \tilde{W}_r) = \sum_{r=l}^{L} \log\left(1 + (r-1)|\beta_r \tilde{W}_r|\right) \; \forall \; r = l, \ldots, L \tag{6.19}$$

where

$$\left(\beta \tilde{W}\right)(r,k) = \sum_{k=1}^{l_{\text{sublayers}}} \beta_{kr} W_{kr} \; \forall \; k = 1, \ldots, l_{\text{sublayers}}, \tag{6.20}$$

$r = l$ is the initial layer from which the penalty is imposed, k is the number of sub-layers, $l_{\text{sublayers}}$ is the number of sub-layer in each layer block and β is one of the profile functions of (6.13) and (6.14).

In the proposed loss function of (6.19), we opt for the logarithm function in order to: *(i)* avoid the β profile functions converging into the constant unity function and *(ii)* prevent gradient explosion, i.e., having the logarithmic decay act as a regularizer [187]. Unlike L_1 norm, the *'log-regularizer'* is non-convex. More broadly, the objective function of an NN is only convex when there are no hidden units, all activations are linear and the design matrix is of full-rank, otherwise, in most cases, the optimisation objective is non-convex [188]. To avoid the challenge of having to design an appropriate transformation [189], it is now standard practice to train such NN designs with the combination of stochastic gradient descent (SGD) and appropriate hyper-parameter tuning. Together with the use of the L_1 norm, these two aspects enforce sparsity in the network weights corresponding to the lowest part of the β profile functions when the regularisation parameter ($\tilde{\lambda}$) is adequately large [184]. In this way, the model size can be scaled down by expunging some layers deterministically during inference, which reduces memory and computational requirements during model deployment with graceful degradation in detection accuracy.

6.5.3 WeSNet with Learnable Weight Profile Coefficients (L-WeSNet)

To improve the robustness of the WeSNet against vanishing gradients and possible gradient explosion, the weight profile functions themselves are made trainable parameters, whose values are optimised during the network training process. This allows for significantly wider exploration of appropriate scaling functions than the predetermined profile functions presented earlier, albeit at the expense of computational complexity during training. To achieve this, (6.9) is modified to include profile weight functions as learned parameters.

$$\tilde{\Psi} = \{\mathbf{W}_{1r},\ \mathbf{W}_{2r}\ ,\ \mathbf{W}_{3r},\ \mathbf{b}_{1r},\ \mathbf{b}_{2r},\ \mathbf{b}_{3r},\ \beta_r\}_{r=1}^{L} \tag{6.21}$$

It is important to note that the monotononicity during training and gradient update is maintained by the shape of the functions of (6.13) and (6.14).

6.6 Complexity Analysis

WeSNet is a truncated version of DetNet, and the detection is performed at the inference layer (see Figure 6.3b) by feed forward computation and subsequent application of the soft sign activation function. The computational cost of WeSNet inference is derived based on the cost of operations of an MLP (please see Appendix B for the details). Our proposed model has 90 layers formed by stacking block of layers, each consisting of three layers DNN. The propagation error is found by computing the derivative of the cost function with respect to the parameters in each block. The computational complexity is specifically measured by the number of operations based on the detector's model. Suppose $\mathbf{A} \in \mathbb{C}^{M\times N}$ and $\mathbf{B} \in \mathbb{C}^{N\times L}$ are arbitrary matrices. $\mathbf{D} \in \mathbb{C}^{M\times N}$ is a diagonal matrix, $\mathbf{a}$, $\mathbf{b} \in \mathbb{C}^{N\times 1}$ and $\mathbf{c} \in \mathbb{C}^{M\times 1}$ are arbitrary vectors and $\mathbf{Q} \in \mathbb{C}^{N\times N}$ is positive definite. The required number of

FLOPs operations of the standard algebraic expressions of interest to this work are summarised in Table 6.1.

Table 6.1: Matrix-vector floating point operations [190]

Expression	Description	Multiplications	Summations	Total Flops
$\alpha\mathbf{a}$	Vector Scaling	N		N
$\alpha\mathbf{A}$	Matrix Scaling	MN		MN
$\mathbf{Ab}$	Matrix-Vector Prod.	MN	$M(N-1)$	$2MN-M$
$\mathbf{AB}$	Matrix-Matrix Prod.	MNL	$ML(N-1)$	$2MNL-ML$
$\mathbf{AD}$	Matrix-Diagonal Prod.	MN		MN
$\mathbf{a}^H\mathbf{b}$	Inner Prod.	N	$N-1$	$2N-1$
$\mathbf{ac}^H$	Outer Prod.	MN		MN
$\mathbf{A}^H\mathbf{A}$	Gram	$\frac{MN(N+1)}{2}$	$\frac{N(M-1)(N+1)}{2}$	$MN^2+N(M-\frac{N}{2})-\frac{N}{2}$
$\lVert\mathbf{A}\rVert_2^2$	Euclidean norm	MN	$MN-1$	$2MN-1$
$\mathbf{Q}^{-1}$	Inverse of Pos. Definite	$\frac{N^3}{2}+\frac{3N^2}{2}$	$\frac{N^3}{2}-\frac{N^2}{2}$	N^3+N^2+N Including N roots

Table 6.2: MIMO detectors' complexity per symbol slot time.

MIMO Detector	Number of Flops Operation
ZF	$\left(\frac{56}{3}\right)N_t^3+38N_t^2+\left(\frac{28}{3}\right)N_t$
MMSE	$\left(\frac{56}{3}\right)N_T^3+40N_t^2+\left(\frac{34}{3}\right)N_t+1$
ML	$\lvert\mathbf{S}\rvert^{N_t}(8N_t^2+8N_t-2)$
SDR	$\left(13N_t^3+25N_t^2+17N_t+4\right)N_{\text{iterations}}$ [18, 106]
WeSNet	$\left[\left(\tilde{\beta}_{cr}N_t(128N_t+5)+9N_t\right)\right]L$, $L=$ number of layers
DetNet	$[(N_t(128N_t-2))]L$

We use the previous equations and the complexity of the feed-forward inference formulation as detailed in Appendix B to compute the number of floating point operations of each MIMO detector. Our results are summarised in Table 6.2, and correspond to the following standard assumptions:

1. One addition, subtraction of a real number is equal to one computational operation.

2. One multiplication of a complex number is equivalent to four real number multiplications and two real number addition.

3. One addition or subtraction of a complex number is equivalent to two real number additions.

4. One division of a complex number is is equivalent to eight real number multiplications and four additions.

Since only a certain fraction of the inputs are used to compute the layer weights of WeSNet and R-WeSNet, most of the operations involved in the feed-forward computations are either sparse vector-matrix multiplication and/or sparse matrix-matrix multiplication. We can evaluate the asymptotic complexity as follows; $\beta_{1r}\mathbf{W}_{1r}$, $\beta_{2r}\mathbf{W}_{2r}$ and $\beta_{3r}\mathbf{W}_{3r}$ for detecting a single received symbol are computed by matrix-vector and matrix-matrix multiplications as $\mathcal{O}\left(\sum_{r=1}^{L}\mathbf{u}_{1r}+\mathbf{s}_{2r}+\mathbf{a}_{3r}\right) = \mathcal{O}(n^3+n^2) = \mathcal{O}(n^3)$.

6.7 Training

First, let us note that training of our model is done once, offline, and can, therefore, accommodate significant complexity followed by the actual deployment of the detector at the inference. Our training dataset comprises transmitted symbols generated stochastically from random normal distribution drawn from either BPSK or 4-QAM constellation, additive white Gaussian noise (AWGN) generated from a uniform distribution over a wide range of SNR values $\mathcal{U}$(8dB - 14dB) and the corresponding received symbols through general random channel taken from a complex Gaussian distribution. On the other hand, our inference (test) dataset, is obtained using the same modulation schemes as the training dataset but with different channel instantiations and distinct instantiations of AWGN over over different range of SNR values $\mathcal{U}$(0dB - 15dB). This training and inference scenario complies with the vast majority of tests in the related literature [22, 23, 117, 122–126]. We train the model for 25000 iterations with 5000 batch size for each iteration on a standard Intel i7-6700 CPU @ 3.40 GHz processor and use Adam Optimiser[127] for gradient descent optimisation. It takes between 17-19 hours to train WeSNet with 20% and 50% profile weight coefficients respectively. This training time is substantial, but it needs to be carried out offline, and only once. As earlier explained, we assume an unknown noise variance during training. We therefore generate the noise vector from a random uniform distribution over the training SNR values $\mathcal{U}(\mathrm{SNR}_{\mathrm{min}}, \mathrm{SNR}_{\mathrm{max}})$. This allows the network to learn over a wide range of SNR conditions.

6.8 Numerical Results

In this section, we present the experimental setup and the performance of the WeSNet under different profile functions and their trainable versions. Amongst deep learning based MIMO detectors, DetNet achieves the best complexity-accuracy performance and also forms the basis of our proposal. Therefore, we deploy and benchmark WeSNet against DetNet, but also present performance comparisons against other classical detectors.

Table 6.3: Simulation settings

Parameters	Values
First Sublayer Dimension	$8N_t = 240$
Second Sublayer Dimension	$N_t = 30$
Third Sublayer Dimension	$2N_t = 60$
Number of Layers	$L = 3N_t = 90$
Fraction of non-zero Layer Weights	$\tilde{\beta}_{\mathrm{cr}}$
Training Samples	500000
Batch Size	5000
Test Samples	50000
Training SNR range	8dB - 14dB
Test SNR range	0dB - 15dB
Optimiser	SGD with Adam
Learning Rate	0.001
Weight Initialiser	Xavier Initialiser
Number of Training Iterations	25000
Number of Monte Carlo during inference	200

6.8.1 Simulation Setup

WeSNet is implemented in *Tensorflow* 1.12.0 [191] using Python. Since deep learning libraries only support real number computations, we use real-valued representation of the random signals and fading channel to generate the training and test datasets. The detector is evaluated under both asymmetric (30 transmit and 60 receive antennas) and symmetric channel (16 transmit and 16 receive antennas) conditions. To ensure a fair comparison with the benchmark model, we use the simulation settings, which are summarised in Table 6.3. The benchmark detectors we consider are:

1. Linear detectors (**ZF** and **MMSE**) implemented based on [19].

2. The optimal detector (**ML**) and optimisation based detector (**SDR**) based on relaxed semidefinite programming as proposed in [105] and [18] respectively.

3. The deep learning-based MIMO detectors **DetNet** as proposed by Samuel *et al.* [22], Samuel *et al.* [117] and **OAMP-Net** introduced by Hengtao *et al.*[23].

6.8.2 Performance of a WeSNet Realisation with Half-Exponential and Linear Profile Functions

For clarity, we begin by defining the following term; WeSNet-(HF/L)-x%: Weight-scaled network obtained from Half-Exponential or Linear profile or function trained and with ‘x’ fraction of the layer weights retained during training and inference.

Figure 6.4 shows the performance of WeSNet with the half-exponential (WeSNet-HF) and linear (WeSNet-L) profile functions of (6.13) and (6.14) when retaining increased percentage of inference layers (as marked in the corresponding legends). The benchmarks comprise DetNet, ZF, MMSE, SDR and ML detectors. Both linear and half-exponential profile WeSNet have comparable performance at lower SNR and profile coefficients between 20% - 30% of the layer weights. As expected, the addition of more profile coefficients increases WeSNet’s detection accuracy, but performance saturates after 60% of the coefficients. However, we observe an appreciable difference at higher SNR as more profile weight coefficients are added. At 10^{-3} BER, WeSNet can be trained with only 10% of the layer weights and still outperforms ZF and MMSE by 1.68 dB and 0.79 dB respectively. Overall, WeSNet with only 20% of the layer weights (WeSNet-HF-20%) achieves virtually the same performance as our benchmark model (DetNet). In fact, with 50% profile weight coefficients (WeSNet-HF-50%), WeSNet outperforms DetNet, producing the accuracy of symbol detection equivalent to SDR. This gain is an experimen-

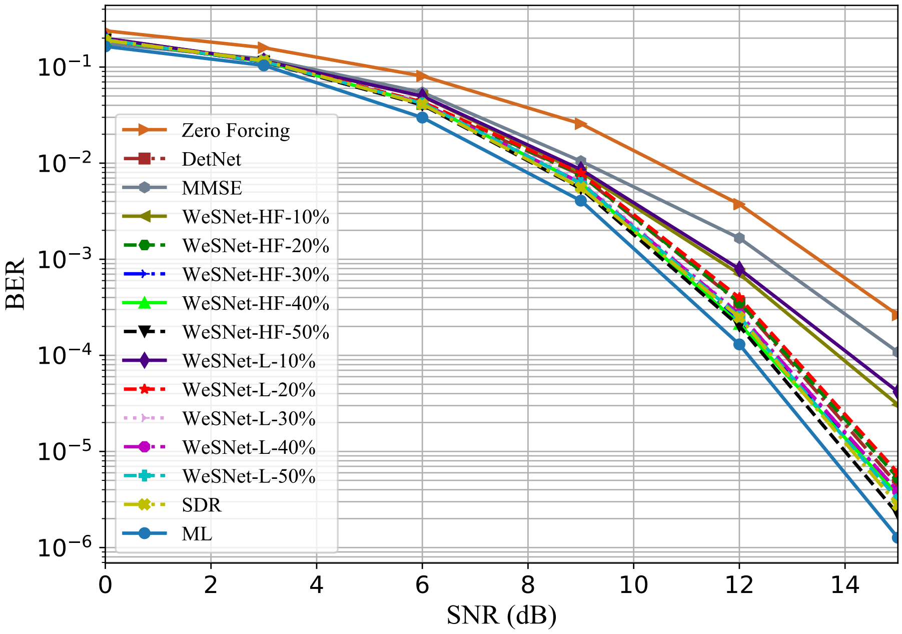


Figure 6.4: BER comparison of the proposed DNN MIMO Detectors (WeSNet-HF, WeSNet-L), DetNet, ZF, MMSE, SDR and ML under 60 × 30 fading channel using BPSK modulation.

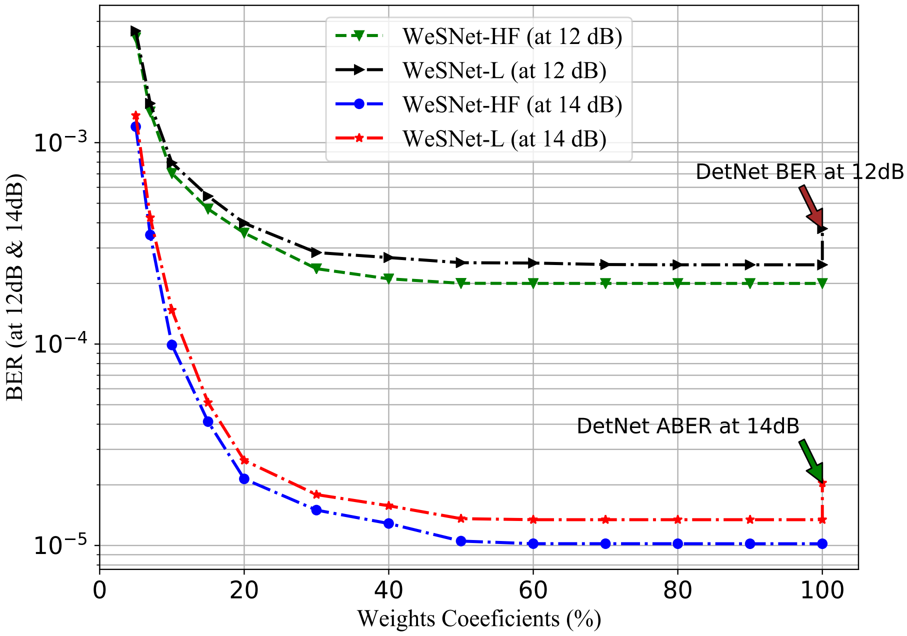
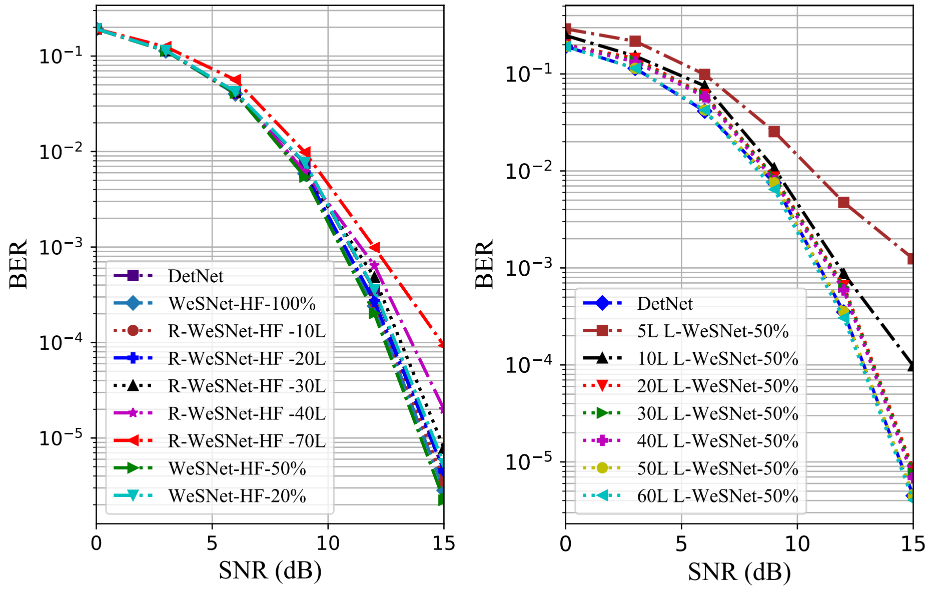


Figure 6.5: BER vs Percentage Weight Profile Coefficients for WeSNet.

tal validation that weight profile functions also act as regularisers, i.e., beyond their sparsity-enforcing property, they also avoid overfitting when the model size grows.

In Figure 6.5, we show the performance of the WeSNet-HF and WeSNet-L



(a) R-WeSNet

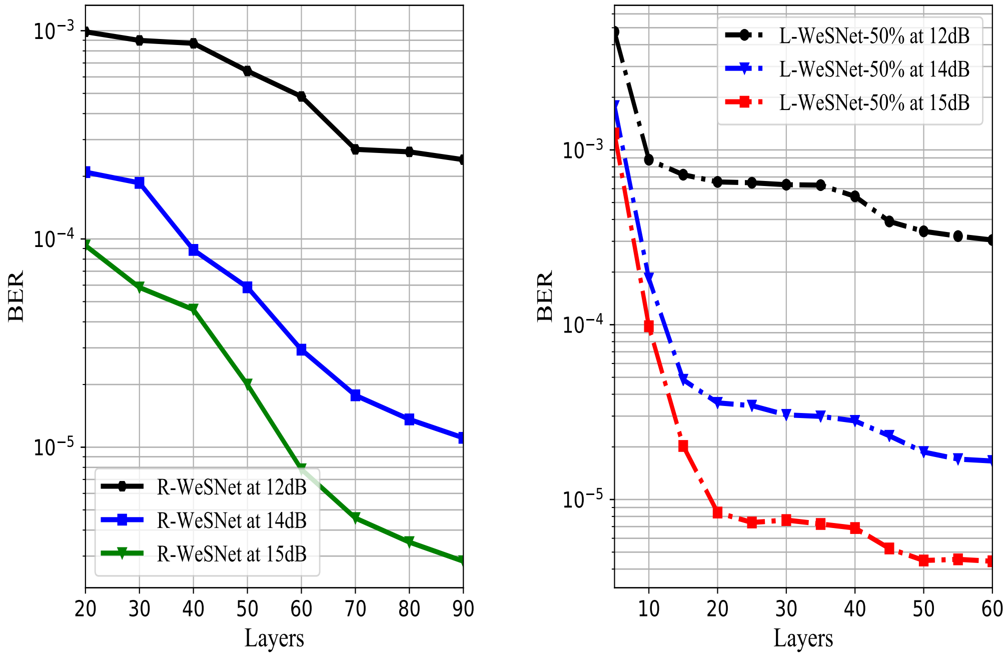
(b) L-WeSNet-50%

Figure 6.6: Performance comparison of R-WeSNet, L-WeSNet trained with 50% profile weight coefficients as a function of layers, WeSNet-HF-50%, WeSNet-HF-20% and DetNet detectors under 60×30 fading channel using BPSK modulation

parametric to the utilised layer weight profile coefficients at 12 dB and 14 dB SNRs. Both outperform DetNet and WeSNet-HF surpasses the WeSNet-L at high SNR. For example, at 14 dB and 10^{-5} BER, it has gain margin of 0.312 dB over the WeSNet-L. We also observe that the BER at 12 dB and 15 dB SNR improves as more profile coefficients are added, but saturates at 50% due to weight saturation. This illustrates that, with the addition of profile weight coefficients, at higher SNR the size of the WeStNet can be scaled down during training by almost 40% - 50% and still achieve almost identical detection accuracy to the full architecture that retains 100% of the weights during training.

6.8.3 Performance Evaluation of R-WeSNet and L-WeSNet

To examine in more detail the performance of our approach against the "direct" approach of removing entire layers by enforcing penalty on the weights through *log*-l_1-norm regularisation, Figure 6.6(a) shows BER-SNR performance curves of the full WeSNet (WeSNet-HF-100%), DetNet and WeSNet when removing entire layers. The figure shows that removing 70 - 40 layers (R-WeSNet-HF-70L and R-



(a) R-WeSNet.

(b) L-WeSNet-50%.

Figure 6.7: BER for R-WeSNet-HF and L-WeSNet vs number of layers

WeSNet-HF-40L) results in considerable loss of accuracy as compared to the corresponding WeSNet-HF models (WeSNet-HF-20% and WeSNet-HF-50%). Nevertheless, 20-30 layers (R-WeSNet-HF-20L and R-WeSNet-HF-30L) can be removed while still achieving BER-SNR performance slightly better than the DetNet's. It can be noticed that a R-WeSNet-HF-10L (with 10 layers short-fall) outperforms both WeSNet-HF-50% and DetNet.

In order to examine the performance of our approach when the weight profile coefficients are made learnable (L-WeSNet), Figure 6.6(b) presents the performance of with 50% learnable weight coefficients (L-WeSNet-HF-50%) over different number of layers. It can be seen that there is a remarkable performance improvement as the size of the network grows from 5 layers to 60 layers. For instance, at 7.2×10^{-2} BER, we observe margin of 2.8 dB between 5 to 30 layers. On the other hand, accuracy remains fairly consistent from 20 to 40 layers. Our study also shows that L-WeSNet-50% produces the same accuracy as DetNet trained with full 90 layers. This means that, for the studied problem, an efficient deep MIMO detector can be designed with 50% trainable weight coefficients and 50 layers.

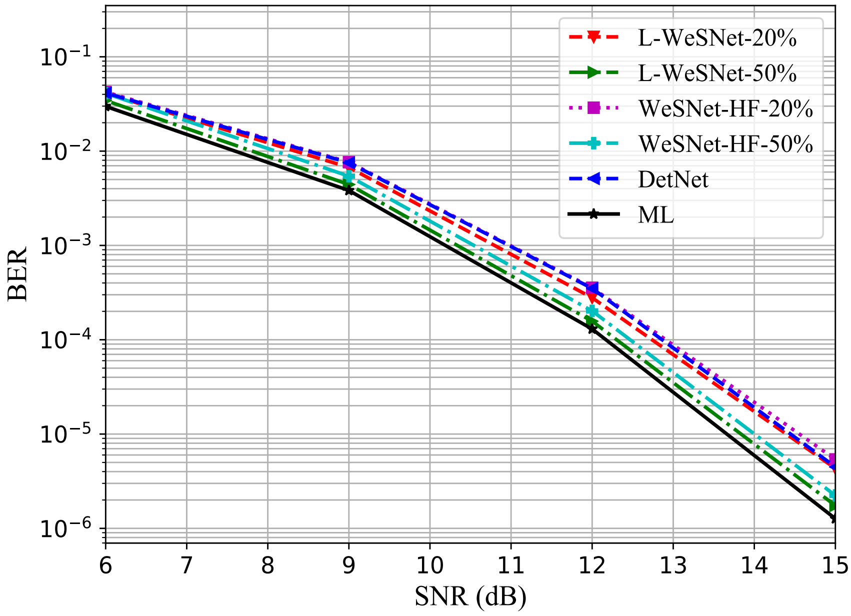


Figure 6.8: Performance comparison of L-WeSNe, WeSNet, DetNet and ML detectors under 60×30 fading channel using BPSK modulation

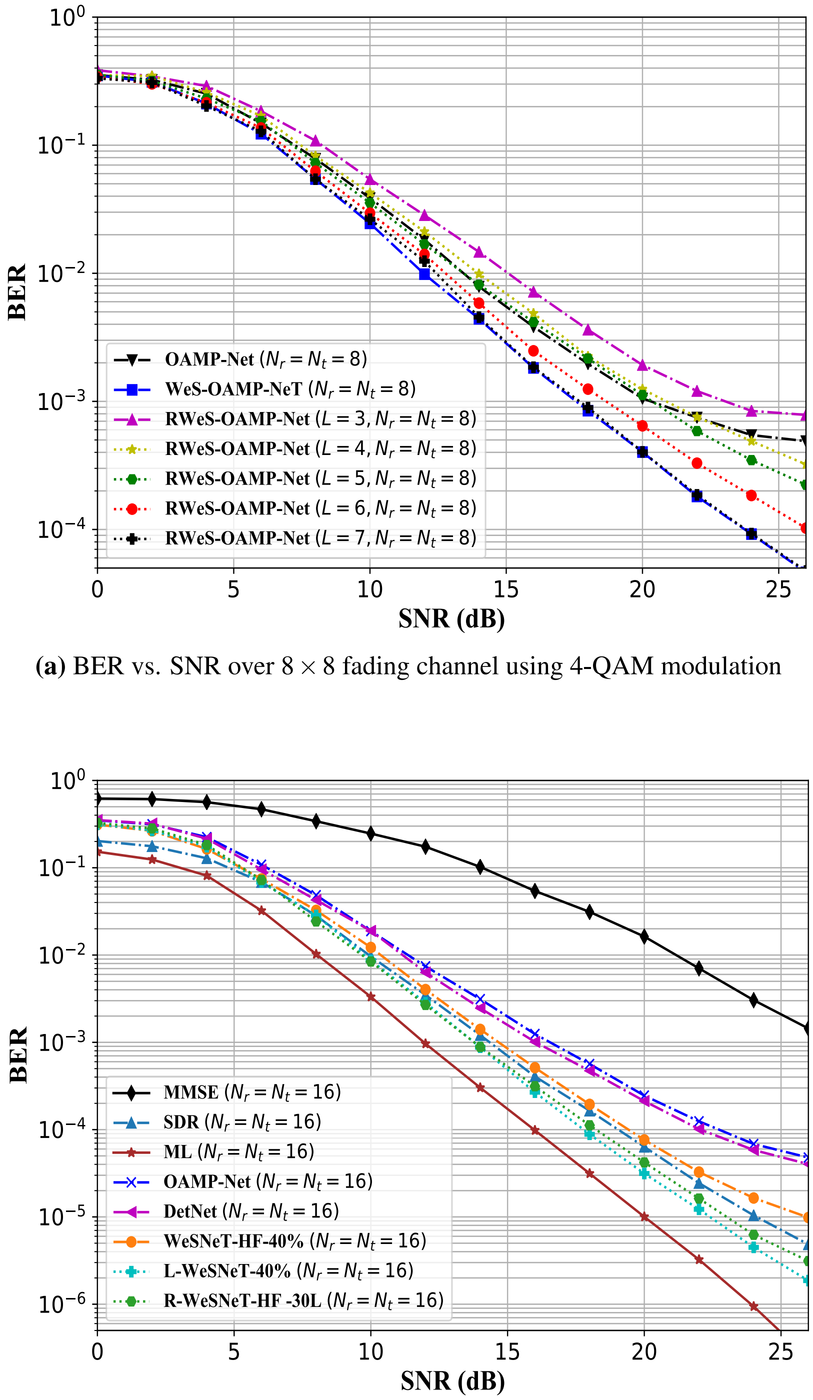
Figure 6.7 shows the average BER for both R-WeSNet and L-WeSNet against the number of layers at 12 dB, 14 dB and 15 dB SNRs respectively. At 12 dB SNR (Figure 6.7(a)), removing 10 - 30 layers during feed forward inference does not significantly change the performance as compared to at 14 dB and 15 dB. However, at 12 dB and 15 dB, we observe a sharp decrease in performance from 70 - 20 layers. The BER is about 10^{-4} at 15 dB and less than 10^{-3} at 14 dB with 40 layers removed. We also see that, at higher SNR, model size can be reduced significantly by removing up to 50 layers during the inference with slightly loss of accuracy.

No rules or analysis exists to precisely determine the size of a neural network (i.e., number of neurons, layers, or layer parameters) for a specific task. Therefore, we train WeSNet-HF with trainable weight coefficients over the different number of layers to determine the conditions under which we can obtain the minimum BER. The average BER is evaluated at 12 dB, 14 dB and 15 dB for each layer configuration as shown in Figure 6.7(b). It can be seen that the BER falls off quickly from 5 - 60 layers. The BER at 14 dB is approximately 3×10^{-5}. This value is reasonably constant from 20 - 30 layers and goes down as more layers are added. It can also be seen that at 15 dB, L-WeSNet-50% produces nearly 10^{-5} BER with only 20 layers.

In Figure 6.8, we compare the BER-SNR performance of WeSNet and L-WeSNet both trained over the entire layers with 20% and 50% of the profile weight coefficients (L-WeSNet-20% and L-WeSNet-50%) against other benchmark models. Our study shows that WeSNet with trainable weight profile functions outperforms the one with non-trainable functions. This comes at the expense of slightly increased training cost due to the additional number of training parameters. This additional training overhead, however, does not increase the inference complexity of the L-WeSNet over WeSNet's, as the inference architectures are the same, except of the difference in the values of the trained weight scaling values. It can be seen that L-WeSNet-20% at 10^{-3} BER outperforms both DetNet and WeSNet-HF-20% by 0.19 dB. Similarly, L-WeSNet-50% yields better detection accuracy over WeSNet-HF-50% and DetNet by 0.22 dB and 0.69 dB, respectively.

6.8.4 Adaptability of WeSNet beyond the DetNet

The proposed approach can be applied to any model that has NN design, including deep unfolding iterative algorithms such as OAMP-Net, TPG-Net, etc. by adding a NN sub-layer design before the estimator and introducing the weight profiling to trade off performance with complexity. Figure 6.9a shows the performance of the OAMP-Net and its weight-scaled version (Wes-OAMP-Net) designed by introducing the weight-scaling framework to OAMP-Net. In addition, we present results with the regularised Wes-OAMP-Net (RWes-OAMP-Net) under scalable reduction of the utilised layers at inference (from $L = 7$ down to $L = 3$), showcasing the scalable accuracy-complexity behaviour of the proposed framework within the OAMP-Net detector. Wes-OAMP-Net (with 10 layers) and the regularised Wes-OAMP-Net (RWes-OAMP-Net) outperform OAMP-Net (with 10 layers). When scaling down complexity at inference, Wes-OAMP-Net with $L = 5$ layers still slightly outperforms conventional OAMP-Net, while the accuracy can be further traded-off for complexity in a graceful manner as more layers are removed during inference. Figure 6.9b shows BER performance of the WeSNet, MMSE, DetNet and OAMP-Net evaluated under symmetric fading channel (16 receive and 16 transmit antennas) using 4-QAM modulation scheme. It can be seen that the performance gap between



(a) BER vs. SNR over 8 × 8 fading channel using 4-QAM modulation

(b) Performance comparison of WeSNet vs benchmark detectors under 16 × 16 channel using 4-QAM modulation

Figure 6.9: Performance evaluation of OAMP-Net vs Weighted-Scaled OAMP-Net (WeS-OAMP-Net)

classical MMSE and ML is significant, i.e., in the range of 15 dB. At lower SNR values (0 - 11 dB), we observe that the performance of WeSNet and OAMP-Net

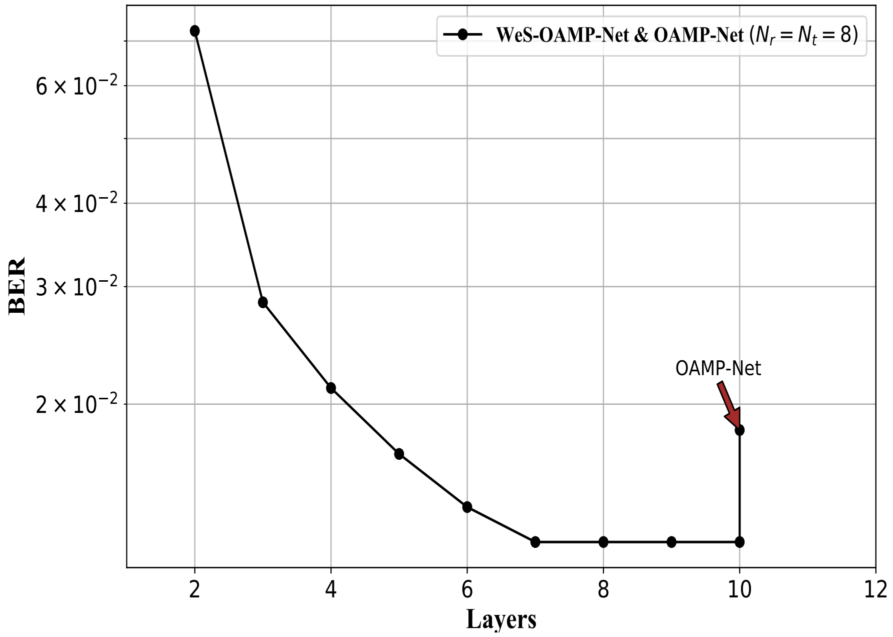
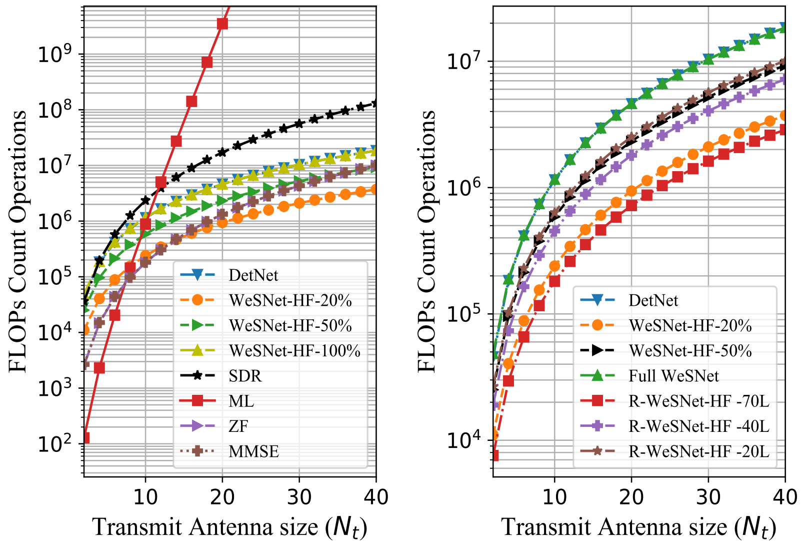


Figure 6.10: BER vs. Layers over a 8 x 8 fading channel using 4QAM modulation.

is the same. However, at higher SNR, DetNet outperforms OAMP-Net slightly in the range of 0.5 dB. We also observe that WesNet-HF-40% and SDR have similar performance, while L-WeSNet-HF-40% outperforms both of them. Similarly, the regularised WeSNet (R-WeSNet-HF) with 30 layers eliminated during the inference is observed to have outperformed all the receivers. Finally, while the DetNet and OAMP-Net have similar performance, WeSNet outperforms both of them at all SNRs, with a reduced (2 - 3 dB) gap to ML.

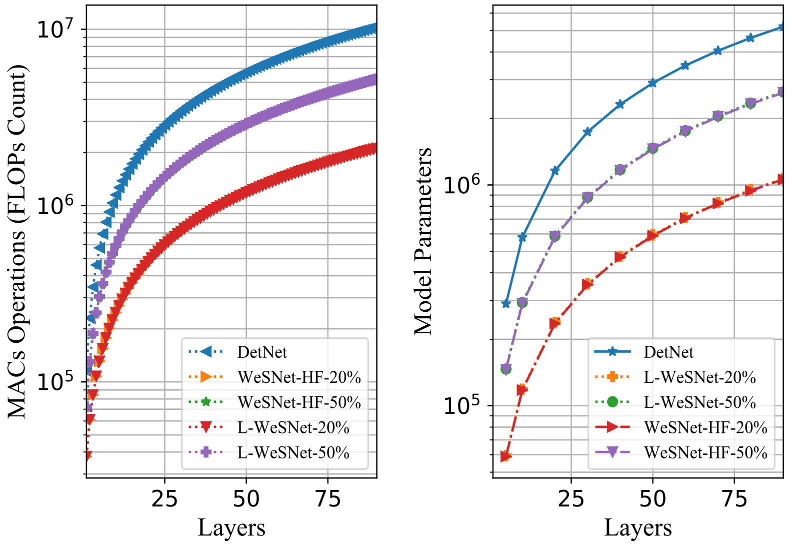
Finally, the performance-complexity tradeoff is further exemplified in Figure 6.10, which shows the BER against number of layers. It can be seen that BER decreases as more layers are added and the BER gains saturate at the seventh layer. We also observe that 2.3 - 41.4% complexity can be saved by reducing the number of OAMP-Net layers from 10 to 3 with loss of accuracy within the range of 0.5 - 3 dB for a system with 8 receive and 8 transmit antennas. Therefore, where before OAMP-Net had only one BER vs complexity operating point in this scenario, our proposed framework has provided a range of BER vs complexity operating points which can be traded-off as per the communication's link requirements.



(a) FLOPs count for WeSNet, DetNet and other classical detectors

(b) FLOPs count of different WeSNet, R-WeSNet models and DetNet

Figure 6.11: Computational complexity comparison of the detectors vs transmit antenna size.



(a) FLOPs count of WeSNet, L-WeSNet and DetNet

(b) Number of parameters of WeSNet, L-WeSNet and DetNet

Figure 6.12: Complexity comparison of WeSNet, L-WeSNet and DetNedt in terms of FLOPs count and model parameters as a function of network layers.

6.8.5 Complexity Evaluation of the Proposed Scheme

To associate layer sizes with complexity and number of antennas in the MIMO configuration, Figure 6.11(a) shows the complexity evaluated as the number of FLOPs for WeSNet-HF-100%, WeSNet-HF-50%, WeSNet-HF-20%, DetNet, ZF, MMSE, SDR and ML detectors against the number of transmit antennas. As expected, as the number of antennas increases, the complexity of ML grows exponentially. On the other hand, WeSNet-HF-20% has the lowest computational cost. As far as the model configuration is concerned, equal number of matrix-matrix and matrix-vector floating point operations are performed by both WeSNet-HF-100% and DetNet during the feed forward inference. However, WeSNet-HF-50% and WeSNet-HF-20% are computationally more efficient than DetNet. For example, with 20% - 80% profile weights coefficients, the training of WeSNet-HF is less complex than that of DetNet under the same operating conditions. When a regularised WeSNet-HF-100% is trained and layers are removed deterministically at inference, Figure 6.11(b) shows that the complexity drops, with graceful degradation in performance. Importantly, as expected from prior experiments, the first 30 layers can be abrogated without any significant compromise on the performance.

Figure 6.12(a) depicts the complexity as function of network layers. The computational requirement grows linearly as more layers are added. It can be observed that the WeSNet-HF-50% and WeSNet-HF-20% are less complex than DetNet over the entire range of layers. Our study shows that, at the inference, the complexity of L-WeSNet is not affected by the presence of learnable weight profile functions. Therefore, WeSNet-HF-50% and WeSNet-HF-20% and their corresponding learnable versions (L-WeSNet-50% and L-WeSNet-20%) have the same computational complexity at inference. Figure 6.12(b) shows the variation of the model size in terms of number of learnable parameters as a function of network layers. For a given layer dimension (number of neurons), the size of the model is determined by the number of layers and the number of trainable parameters. It can be seen that WeSNet, in addition to having better detection accuracy, it is substantially more memory efficient than DetNet and requires less training time under the same experimental conditions.

As more weights profile coefficients are added, the number of FLOPs increases. Figure 6.13 shows how the computational cost and model parameters change with the profile weight coefficients. As shown by earlier experiments,

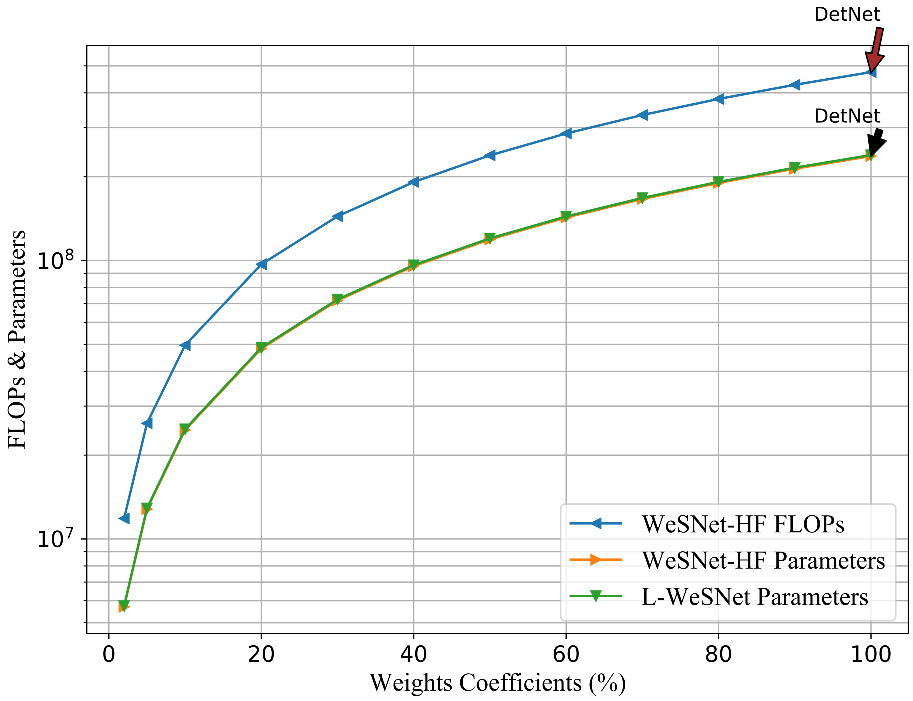


Figure 6.13: Total FLOPs and model parameters vs weight profile coefficients.

WeSNet-HF achieves performance close to DetNet with only 20% to 30% of the layer weights. Therefore, such weight scaling leads to a significant decrease in the model size by 79.82% and 68.73% respectively. Similarly, we observe a reduction of 51.43% computational cost and 49.78% decrease in model size with 50% profile weight coefficients.

6.9 Summary

In this chapter, we present an efficient and scalable deep neural network based MIMO detector, where complexity can be adjusted at inference with graceful degradation in the detection accuracy. We introduce a weight scaling framework using monotonically non-increasing profile functions to dynamically prioritise a fraction of the layer weights to be used during training. In order to allow for the neural network architecture to self-adjust to the detection complexity, we also allow for the profile functions themselves to be trainable parameters in the proposed architecture. From our simulation results, we find that the model with trainable coefficients outperforms the one with non-trainable coefficients, but at the cost of complexity.

In addition, our proposal shows that adding weight scaling via monotonic profile functions maintains the detection accuracy when dropping layer weights. This is achieved in part by introducing an $log - l_1$-norm based regularisation function on the layer weights and their profile function coefficients so that the model size can be scaled down by nearly 40% during the feed-forward inference with marginal impact in the detection accuracy.

Chapter 7

Concluding Remarks and Future Work

Downlink transmit precoding and signal detection techniques are quintessential requirements for exploiting the benefits of spatial multiplexing of multiple-antenna systems. Considering the hardware requirements for practical implementation and deployment, integrating ML techniques for downlink transmission designs is needed for future 5G and beyond wireless communication systems. Correspondingly, this thesis studies model and data-driven ML approaches and proposes various scalable memory-efficient and hardware-inspired DL frameworks for multiuser downlink transmission strategies and signal MIMO detection designs.

7.1 Summary and Conclusion

Chapter 2 of this thesis presents a general review of fundamental theoretical concepts of MIMO communication systems and the related technologies. We have focused explicitly on the downlink transmission and reviewed different linear and nonlinear precoding and detection techniques that have been reported in the literature. The chapter highlights the merit and demerits of such signal processing schemes. Furthermore, the chapter reviews various existing precoding optimisation schemes based on the conventional approach, where the MUI interference is treated as a pernicious entity. The optimisation-based precoding schemes, where the MUI is exploited and combined with the knowledge of data symbols at the BS for QPSK and 8PSK modulated signals, are also reviewed. Similarly, the chapter has presented a detailed rundown of the MIMO detectors based on the iterative algorithms reported in the literature and their corresponding unfolded DL-based counterparts. Additionally, Chapter 3 presents a recap of relevant ML theoretical foundations. Distinctively, the chapter expounds on ML algorithms and their applications in wireless communications. An overview of DL-based MIMO detectors derived from traditional iterative algorithms is also presented.

Following the above two introductory chapters that explain the fundamental concepts and literature review exclusively relevant to this research work, the details of the main contributions of this thesis are presented in Chapter 4 - Chapter 6. More specifically:

- Chapter 4 investigates applications of interference exploitation to an MU-MIMO downlink transmission system with ML. We propose a novel unsupervised learning-based precoding framework that trains DNNs with no target labels by unfolding an interior point method (IPM) proximal *'log'* barrier function. Different proximal *'log'* barrier functions are derived based on strict and relaxed power minimisation formulations subject to SINR constraints. The proposed scheme exploits the known interference via SLP to minimise the downlink transmit power. The idea is also extended to robust power minimisation problem, where channel error due to uncertainty in the channel estimation is considered. The main observations of this chapter are elucidated as follows:

    1. Thanks to IPM SLP proximal 'log' barrier function, the performance of the proposed SLP-DNet is promising commensurate with the SLP optimisation-based solutions. With this approach, we use the original SLP optimisation Lagrange function as a loss function with an additional regularisation term. When there is not sufficient data to train the model, the proposed learning scheme is attractive because it opens a way of transforming constrained optimisation problems into an unfolded sequence of unconstrained subproblems that can be trained in an unsuper-

vised manner.

2. The gain in the transmit power reduction of the proposed SLP-DNet precoding schemes is near the optimal SLP optimisation-based precoding method. The loss of performance is worth the benefits of the reduced computational complexity offered by the proposed unsupervised learning-based precoding solutions. Therefore, our proposals demonstrate an indispensable balance between the performance and the computational complexity involved. However, we observe that transforming the optimisation problem into a learning framework is increasingly challenging as the number of constraints increases (see robust SLP formulation), translating into additional computational complexity, as clearly shown in the execution time.

3. The proposed SLP-SNet is feasible for all BS antennas and mobile user configurations. More importantly, we observe that the performance gap between the SLP optimisation-based and proposed learning-based techniques closes rapidly as more users are served. This observation further highlights the flexibility of the proposed SLP-DNet and the possibility of extending it to a multi-cell scenario.

- To reduce the offline and online computational requirements of the learning frameworks developed in Chapter 4, we introduce the concept of NN model compression in Chapter 5. The compression is performed through NN weights quantisation, where the weights are quantised to binary $(-1,+1)$ and ternary $(-1,0,+1)$ values to reduce the computational complexity of the developed learning architectures. Fully quantised SLP-DNet models (SLP-DBNet and SLP-DTNet) offer complexity reduction gains in terms of computational inference power between $40\% - 58\%$ and memory savings up to $\sim 21\times$ and $\sim 13\times$ compared to the full-precision SLP-DNet, respectively. However, they suffer a performance loss compared to the conventional SLP optimisation-based solutions. We propose a stochastic quantisation based on binary and ternary quantisations (SLP-DSQBNet and SLP-DSQTNet) to address this drawback. With stochastic quantisation, part of the NN weight matrix is quantised to either binary or ternary, and the remaining portion is retained in complete floating-point numerical precision. A lottery disc-like algorithm combined with a monotonically non-increasing probability function for selecting the row of the NN channel/filter weights to be quantised is introduced. The main observations of this chapter are:

1. The introduction of quantisation within the NN weight tensor has a significant impact on the performance of the precoder. Specifically, the proposed SLP-DBNet and SLP-DTNet incur a substantial performance loss due to the non-homogeneous nature of the quantisation error at each iteration, leading to a lousy gradient direction during training. In addition to significant memory savings offered by SLP-DBNet and SLP-DTNet, they show corresponding computational complexity reductions of $\sim 20\times$ and $\sim 10\times$, respectively, compared to plane SLP-DNet. To improve the performance further, we propose a stochastic quantisation technique. Here, the quantisation error is used to direct the gradient descent towards the best local minima during training, improving the performances of SLP-DSQBNet and SLP-DSQTNet compared to their fully quantised counterparts. While SLP-DSQBNet and SLP-DSQTNet exhibit promising performances, SLP-DSQT, in particular, offers higher power savings of $50\% - 58\%$ comparable to that of the optimal solution.
2. A considerable transmit power increase is observed where the channel uncertainty lies within the region of CSI error bounds of $\varsigma^2 = 10^{-3}$. Interestingly, like the robust SLP optimisation-based scheme, the proposed quantised DN-based SLP models show a descent transmit power savings by exploiting the CI. This observation reveals the potential benefits of exploiting quantisation techniques to build learning-based precoders instead of adopting traditional fully precisioned DNN models.
3. The proposed DNN quantisation is promising for online inference and the realistic implementation of learning-based precoders and signal de-

tectors on practical communication systems. It may be exceptionally effective for 5G and beyond communication systems that require many antenna arrays at the BS, potentially leading to the decreased complexity of signal processing problems involved.

- In Chapter 6, we study downlink MIMO detection strategies for m-MIMO systems. Unequivocally, this chapter focuses on general systematic structural simplification by dynamically scaling the NN weight values using monotonically non-increasing functions to design efficient learning-based MIMO detectors. The proposed concept is applied to the state-of-the-art DL-based MIMO detector, DetNet. Despite the performance of the DetNet, its heuristic nature makes its NN design challenging to understand and modify. Because DetNet is the first learning-based MIMO detector whose performance matches the SDR MIMO detector, it presents the best baseline model for learning-based decoders. To address this challenge, we propose a systematic NN weight scaling mechanism to improve network performance over a wide range of signal modulation schemes and significantly reduce unnecessary model complexity. We extend the idea to the learning-based iterative algorithms that do not have an explicit NN design, such as OAMP-Net, to improve their expressibility. The following are the critical remarks based on results observed in this chapter:

    1. The introduction of non-increasing monotonic profile functions allows us to modify the structure of the DetNet and can be applied to any learning model that contains an NN architecture. The two functions employed are linear and half-exponential and multiplied element-wise across the NN weight's elements. We observe that these modifications allow us to dynamically cut down the size of the DetNet and improve its learning ability with a significant complexity reduction in terms of the number of learning parameters, training and inference times. The proposed NN scaling approach presents a potential generic structural simplification mechanism for reducing model computational complex-

ity. It will be interesting to explore different types of non-decreasing monotonic profile functions.

2. We have exploited the *log* l_1-norm penalty function to induce sparsity to the proposed learning-based MIMO detection network by adding a regularisation term to allow the network to adjust its training strategy so that some layers can be dynamically exterminated during inference without compromising the performance. The regularised WeSNet with fewer layers achieves impressive performance better than the model with many cascade layers. The approach is particularly effective when the dimension of the dataset is large, or the number of the transmit antennas is large.

7.2 Possible Future Extensions

In this thesis, we have investigated the concept of model-driven and data-driven DL and developed specific learning frameworks based on expert or domain knowledge by transforming the original constrained and unconstrained optimisation problems into learning layers of NN architectures for SLP and MIMO detector designs. We have extensively studied lightweight NN methods through model structural simplification and weight quantisation to design fast and compressed DL-based architectures for SLP and MIMO detection. While this work has fully covered some physical layer communications areas, others remain unexplored. Our studies have sparked off further investigations in the following directions as follows:

- **General learning framework design:** The proposed DL approaches can potentially open a new way of developing generic and customised learning architectures based on the specific optimisation problems for physical layer communications. For instance, this idea can be extended to SINR balancing problems, sum-rate maximisation problems, interference alignment, network management, modulation classification, etc. Furthermore, the adaptability of the proposed learning schemes should be studied when tested in different environmental settings, e.g. testing the model with a channel type different from the one it was trained.

- **DL frameworks for constrained optimisation problems:** Large antenna arrays provide phenomenal performance vis-a-vis transmission reliability and high data rate communications. It offers enormous amounts of baseband data, which can be used to assess the environment. The ability of DL algorithms to deal with data makes them suitable to analyse the vast amounts of data generated by m-MIMO arrays. Because parametric models are usually complex, classical signal processing, such as precoding, detection or equalisation schemes, require iterative algorithms that are challenging to run on practical systems. More specifically, in terms of function approximation and iterative algorithmic approaches, deep-unfolding offers the most promising path to new learning-based designs for physical layer communications problems. Significant future research directions should include both relevant physical modelling and the development of an algorithmic framework that exploits appropriate ML tools and convexity of the constrained optimisation problems. While trained DNNs represent an indispensable technology in this context, other learning methods, such as dictionary learning techniques involving a learning matrix as a sparse signal, are also crucial for designing scalable models. In addition, it will reduce the computational overheads of online training, which are often associated with conventional model-based signal processing techniques. The model trained with the synthetically generated channel instantiation should be validated with experimental data obtained from real measurements. Such a practice will outstandingly facilitate state-of-the-art learning frameworks for 5G and beyond technologies.

- **Combined NN weight scaling and quantisation:** In Chapter **5**, we have considered DNN compression via quantisation, where a probability function is used to select the portion of the weight matrix to be quantised based on the magnitude of the quantisation error. It will be an exciting feature work to combine the approaches described in Chapter **5** and Chapter **6**. Instead of quantising the portion of the weights based on quantisation error, the least significant NN weights, which are dynamically removed, as described in Chapter **6**, can be quantised. This may potentially deal with the impact of the quantisation error on training convergence and model learning instability.

- **Running trained model on different set of users and dimension scaling:** Throughout the thesis, we have considered a BS with a fixed number of antennas serving different simultaneous users. For a given BS, DNN models trained for a specific number of users cannot be used for another set of users due to the change in the dimension of the input dataset. The model has to be trained for every new set of users. A compelling future work will be to develop a generic conversion module that will deal with the variation of the input matrix shapes and allow the trained model to be tested without necessarily retraining it from scratch for every new transmit-receive antenna configuration. Furthermore, the input data dimension is often a multiple of the transmit-receiver composition or channel matrix dimension. However, as for the intermediate layers, the dimensions are arbitrarily selected. Depending on the size of the problem, scaling these dimensions can affect the accuracy of the trained model. Unfortunately, the traditional scaling process has not yet been fully understood. The conventional way of doing it is arbitrarily scaling across depth, width and resolution (channel) in the case of CNN, which requires manual tuning. It will be a mesmerising future research direction to consider the model dimension optimisation, where the model accuracy (performance) is maximised by finding the appropriate layer dimensions subject to given target memory and computational target cost in FLOPs.

- **Extension to MU-MIMO SLP:** The proposed learning-based SLP can be extended to a multiuser system with multiple receive antennas (MU-MIMO system), where multiple independent radio terminals are enabled to access a system, enhancing the communication capabilities of each terminal. Multiple users can simultaneously access the same channel to exploit the maximum system capacity offered by MIMO's spatial degrees of freedom. While such antenna configuration provides higher throughput, it will come with additional hardware and computational costs on both BS and receiver sides. Therefore, scalable DNN frameworks proposed in this thesis will facilitate the designs of memory-efficient and low computationally efficient learning-based models for MU-MIMO SLP designs to address such complexities.

Conclusively, this thesis has presented potential machine learning applications, specifically DL, in physical layer communications. Several strategies for designing low complexity DNN frameworks for interference management at the BS via pre-coding and receiver designs have been developed. The author desires the solutions, results, and conclusions stemmed within this thesis will help explore the potentials and invigorate new novel ML strategies for the 5G and beyond wireless communication systems.

Appendix A

Proximity operator barrier for Robust SLP

For every transmit precoding vector $\mathbf{w}_2 \in \mathbb{R}^{2N_t \times 1}$, the proximity operator of the barrier $\gamma\mu B_{\text{robust}}(\mathbf{w}_2)$ is given by

$$\Phi_{\text{rb}}(\mathbf{w}_2, \gamma, \mu) = \frac{2\Gamma n_0 \tan^2\phi - X(\mathbf{w}_2, \gamma, \mu)^2}{2\Gamma n_0 \tan^2\phi - X(\mathbf{w}_2, \gamma, \mu)^2 + 2\gamma\mu} \mathbf{w}_2 \tag{A.1}$$

where $X(\mathbf{w}_2, \gamma, \mu)$ is the unique solution of the cubic equation expressed as [159]

$$x^3 - \left(\left(\varsigma^2 - \hat{\Phi}^T \hat{\Phi}\right) \|\mathbf{w}_2\|_2 + 4\hat{\Phi}^T \mathbf{w}_2 \tan\phi \sqrt{\Gamma n_0}\right) x^2 + \left(2\Gamma n_0 \tan^2\phi + 2\gamma\mu\right) x + \\ 2\Gamma n_0 \tan^2\phi \left(\left(\varsigma^2 - \hat{\Phi}^T \hat{\Phi}\right) \|\mathbf{w}_2\|_2 + 4\hat{\Phi}^T \mathbf{w}_2 \tan\phi \sqrt{\Gamma n_0}\right) = 0. \tag{A.2}$$

where $\hat{\Phi} = \Phi^T \Upsilon$. It can be observed that (A.2) is a cubic equation and can be solved analytically. In the final analysis, following similar steps as in (4.23)-(4.27), the robust deep-unfolded model is obtained by finding the Jacobean matrix of (A.1) with respect to the optimisation variable $\mathbf{w}_2$, and the derivatives with respect to the step-size $\gamma > 0$ and the Lagrange multiplier associated with the inequality constraint $\mu > 0$. We use similar concepts presented in subsection 4.3.3 to formulate the learning algorithm of the robust SLP as a series of sub-problems with respect to the combined effect of the two inequality constraints as follows

$$\min_{\mathbf{w_2} \in \mathbb{R}^{\mathbf{2N_T} \times \mathbf{1}}} \quad \|\mathbf{w}_2\|_2^2 + \lambda \mathbf{w}_2 + \mu B_{\text{robust}}(\mathbf{w}_2). \tag{A.3}$$

Similar to a nonrobust SLP-DNet, the update rule for every iteration is expressed as

$$\mathbf{w}_2^{[r+1]} = \text{prox}_{\gamma^{[r]}\mu^{[r]}B_{\text{robust}}}\left(\mathbf{w}_2^{[r]} - \gamma^{[r]}\Delta D_{\text{robust}}(\mathbf{w}_2^{[r]}, \lambda^{[r]})\right) \tag{A.4}$$

where

$$D_{\text{robust}}(\mathbf{w}_2^{[r]}, \lambda^{[r]}) = \|\mathbf{w}_2\|_2^2 + \lambda \mathbf{w}_2. \tag{A.5}$$

Appendix B

Feed-Forward Computational Cost of an MLP

Consider an input, $\mathbf{X} \in \mathbb{R}^{(j,k)}$ and weight $\mathbf{W} \in \mathbb{R}^{(i,j)}$, the linear combination of $\mathbf{X}$ and $\mathbf{W}$ is given by

$$\mathbf{Z}_{ik} = \mathbf{W}_{ij}\mathbf{X}_{j,k} + \mathbf{b}_i \tag{B.1}$$

Applying non linear activation to (B.1), gives:

$$\mathbf{a}_{ik} = g(\mathbf{Z}_{ik}) \tag{B.2}$$

where $g(\cdot)$ is the nonlinear activation function. The matrix multiplication has an asymptotic computational complexity $\mathscr{O}(n^3)$ and the activation function has $\mathscr{O}(n)$ complexity.

B.1 Feed-Forward Inference

For $N^{[L]}$ number of neurons including bias unit in the r-th layer, the total complexity can be calculated as a sum of the total number of matrix multiplication and the applied activation over the entire layers as

$$N_{\text{matmul}} = \sum_{r=2}^{L} (N^{[r]}N^{[r-1]}N^{[r-2]}) + N^{[1]}N^{[0]} \tag{B.3}$$

$$N_g = \sum_{r=1}^{L} (N)^{[r]} \tag{B.4}$$

$$\begin{aligned} \text{Complexity} &= N_{\text{matxmul}} + N_g \\ &= N_L \ . \ N^3 \end{aligned} \tag{B.5}$$

The complexity for r-th layers:

$$\begin{aligned} N_{\text{matmul}} &= \mathcal{O}(n \ . \ n^3) \\ &= \mathcal{O}(n^4) \end{aligned} \tag{B.6}$$

Similarly, the complexity N_g for the activation function with L layers is:

$$\begin{aligned} N_g &= N_L \ . \ N \\ &= \mathcal{O}(n^2) \end{aligned} \tag{B.7}$$

Therefore, the total complexity of the forward propagation is

$$\begin{aligned} \text{Total complexity} &= \mathcal{O}(n^4 + n^2) \\ &\approx \mathcal{O}(n^4) \end{aligned} \tag{B.8}$$

Bibliography

[1] Hyogi Jung. Cisco visual networking index: global mobile data traffic forecast update 2010–2015. Technical report, Technical Report, Cisco Systems Inc. 2011. Available online: https://www . . . , 2011.

[2] Shaoshi Yang and Lajos Hanzo. Fifty years of MIMO Detection: The road to Large-scale MIMOs. *IEEE Communications Surveys & Tutorials*, 17(4):1941–1988, 2015.

[3] Jakob Hoydis, Stephan Ten Brink, and Mérouane Debbah. Massive MIMO in the UL/DL of cellular networks: How many antennas do we need? *IEEE Journal on selected Areas in Communications*, 31(2):160–171, 2013.

[4] Lu Lu, Geoffrey Ye Li, A Lee Swindlehurst, Alexei Ashikhmin, and Rui Zhang. An overview of massive mimo: Benefits and challenges. *IEEE journal of selected topics in signal processing*, 8(5):742–758, 2014.

[5] Nusrat Fatema, Guang Hua, Yong Xiang, Dezhong Peng, and Iynkaran Natgunanathan. Massive MIMO linear precoding: A survey. *IEEE systems journal*, 12(4):3920–3931, 2017.

[6] Cheng Zhang, Yindi Jing, Yongming Huang, and Luxi Yang. Performance analysis for massive MIMO downlink with low complexity approximate zero-forcing precoding. *IEEE Transactions on Communications*, 66(9):3848–3864, 2018.

[7] Woon Hau Chin, Zhong Fan, and Russell Haines. Emerging technologies

and research challenges for 5G wireless networks. *IEEE Wireless Communications*, 21(2):106–112, 2014.

[8] David Garcia-Roger, Edgar E González, David Martín-Sacristán, and Jose F Monserrat. V2X support in 3GPP specifications: From 4G to 5G and beyond. *IEEE Access*, 8:190946–190963, 2020.

[9] Rubayet Shafin, Lingjia Liu, Vikram Chandrasekhar, Hao Chen, Jeffrey Reed, et al. Artificial intelligence-enabled cellular networks: A critical path to beyond-5G and 6G. *arXiv preprint arXiv:1907.07862*, 2019.

[10] Manuel Eugenio Morocho-Cayamcela, Haeyoung Lee, and Wansu Lim. Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions. *IEEE Access*, 7:137184–137206, 2019.

[11] Hengtao He, Shi Jin, Chao-Kai Wen, Feifei Gao, Geoffrey Ye Li, and Zongben Xu. Model-driven deep learning for physical layer communications. *IEEE Wireless Communications*, 26(5):77–83, 2019.

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

[13] Cheng-Xiang Wang, Marco Di Renzo, Slawomir Stanczak, Sen Wang, and Erik G Larsson. Artificial intelligence enabled wireless networking for 5G and beyond: Recent advances and future challenges. *IEEE Wireless Communications*, 27(1):16–23, 2020.

[14] Maha Alodeh, Symeon Chatzinotas, and Björn Ottersten. Data aware user selection in cognitive downlink MISO precoding systems. In *IEEE International Symposium on Signal Processing and Information Technology*, pages 000356–000361. IEEE, 2013.

[15] Maha Alodeh, Symeon Chatzinotas, and Björn Ottersten. A multicast approach for constructive interference precoding in MISO downlink channel.

In *2014 IEEE International Symposium on Information Theory*, pages 2534–2538. IEEE, 2014.

[16] Ka Lung Law and Christos Masouros. Constructive interference exploitation for downlink beamforming based on noise robustness and outage probability. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3291–3295. IEEE, 2016.

[17] Nicholas D Sidiropoulos and Z-Q Luo. A semidefinite relaxation approach to MIMO detection for high-order qam constellations. *IEEE signal processing letters*, 13(9):525–528, 2006.

[18] Wing-Kin Ma, Chao-Cheng Su, Joakim Jaldén, and Chong-Yung Chi. Some results on 16-QAM MIMO detection using semidefinite relaxation. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2673–2676. IEEE, 2008.

[19] Cheng-Yu Hung and Wei-Ho Chung. An improved MMSE-based MIMO detection using low-complexity constellation search. In *2010 IEEE Globecom Workshops*, pages 746–750. IEEE, 2010.

[20] Charles Jeon, Ramina Ghods, Arian Maleki, and Christoph Studer. Optimality of large MIMO detection via approximate message passing. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 1227–1231. IEEE, 2015.

[21] Ang Li and Christos Masouros. Interference exploitation precoding made practical: Optimal closed-form solutions for psk modulations. *IEEE Transactions on Wireless Communications*, 17(11):7661–7676, 2018.

[22] Neev Samuel, Tzvi Diskin, and Ami Wiesel. Deep MIMO detection. In *Signal Processing Advances in Wireless Communications (SPAWC), 2017 IEEE 18th International Workshop on*, pages 1–5. IEEE, 2017.

[23] Hengtao He, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li. A model-driven deep learning network for MIMO detection. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 584–588. IEEE, 2018.

[24] Wenchao Xia, Gan Zheng, Yongxu Zhu, Jun Zhang, Jiangzhou Wang, and Athina P Petropulu. A deep learning framework for optimization of miso downlink beamforming. *IEEE Transactions on Communications*, 68(3):1866–1880, 2019.

[25] Hongji Huang, Yiwei Song, Jie Yang, Guan Gui, and Fumiyuki Adachi. Deep-learning-based millimeter-wave massive MIMO for hybrid precoding. *IEEE Transactions on Vehicular Technology*, 68(3):3027–3032, 2019.

[26] Ziyue Lei, Xuewen Liao, Zhenzhen Gao, and Ang Li. CI-NN: A model-driven deep learning based constructive interference precoding scheme. *IEEE Communications Letters*, 2021.

[27] Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nicholas D Sidiropoulos. Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing*, 66(20):5438–5453, 2018.

[28] Ahmed Alkhateeb, Sam Alex, Paul Varkey, Ying Li, Qi Qu, and Djordje Tujkovic. Deep learning coordinated beamforming for highly-mobile millimeter wave systems. *IEEE Access*, 6:37328–37348, 2018.

[29] Paul de Kerret and David Gesbert. Robust decentralized joint precoding using team deep neural network. In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–5. IEEE, 2018.

[30] Hao Huang, Yang Peng, Jie Yang, Wenchao Xia, and Guan Gui. Fast beamforming design via deep learning. *IEEE Transactions on Vehicular Technology*, 69(1):1065–1069, 2019.

[31] David Gesbert, Mansoor Shafi, Da-shan Shiu, Peter J Smith, and Ayman Naguib. From theory to practice: An overview of MIMO Space-time Coded Wireless Systems. *IEEE Journal on selected areas in Communications*, 21(3):281–302, 2003.

[32] Lizhong Zheng and David N. C. Tse. Diversity and Multiplexing: A fundamental tradeoff in Multiple-antenna Channels. *IEEE Transactions on information theory*, 49(5):1073–1096, 2003.

[33] Arogyaswami J Paulraj, Dhananjay A Gore, Rohit U Nabar, and Helmut Bolcskei. An overview of MIMO communications-a key to gigabit Wireless. *Proceedings of the IEEE*, 92(2):198–218, 2004.

[34] Yong Soo Cho, Jaekwon Kim, Won Y Yang, and Chung G Kang. *MIMO-OFDM Wireless Communications with MATLAB*. John Wiley & Sons, 2010.

[35] Omar El Ayach, Steven W Peters, and Robert W Heath. The Practical Challenges of Interference Alignment. *IEEE Wireless Communications*, 20(1):35–42, 2013.

[36] Robert W Heath and Arogyaswami J Paulraj. Switching between Diversity and Multiplexing in MIMO Systems. *IEEE Transactions on Communications*, 53(6):962–968, 2005.

[37] Richard B Ertel, Paulo Cardieri, Kevin W Sowerby, Theodore S Rappaport, and Jeffrey H Reed. Overview of spatial channel models for antenna array communication systems. *IEEE personal communications*, 5(1):10–22, 1998.

[38] Martin Steinbauer, Andreas F Molisch, and Ernst Bonek. The double-directional radio channel. *IEEE Antennas and propagation Magazine*, 43(4):51–63, 2001.

[39] Da-Shan Shiu, Gerard J Foschini, Michael J Gans, and Joseph M Kahn. Fading correlation and its effect on the capacity of multielement antenna systems. *IEEE Transactions on communications*, 48(3):502–513, 2000.

[40] Brijesh Kumbhani and Rakhesh Singh Kshetrimayum. *MIMO wireless communications over generalized fading channels*. CRC Press, 2017.

[41] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[42] Mai Vu and Arogyaswami Paulraj. MIMO wireless linear precoding. *IEEE Signal Processing Magazine*, 24(5):86–105, 2007.

[43] Christoph Windpassinger, Robert FH Fischer, Tomas Vencel, and Johannes B Huber. Precoding in multiantenna and multiuser communications. *IEEE Transactions on Wireless Communications*, 3(4):1305–1316, 2004.

[44] Michel T Ivrlac, Wolfgang Utschick, and Josef A Nossek. Fading correlations in wireless MIMO communication systems. *IEEE Journal on selected areas in communications*, 21(5):819–828, 2003.

[45] Christos Masouros, Mathini Sellathurai, and Tharm Ratnarajah. Large-scale MIMO transmitters in fixed physical spaces: The effect of transmit correlation and mutual coupling. *IEEE Transactions on Communications*, 61(7):2794–2804, 2013.

[46] Michael Botros Shenouda and Timothy N Davidson. On the design of linear transceivers for multiuser systems with channel uncertainty. *IEEE Journal on Selected Areas in Communications*, 26(6):1015–1024, 2008.

[47] Qian Zhang, Chen He, and Lingge Jiang. Per-stream MSE based linear transceiver design for MIMO interference channels with CSI error. *IEEE Transactions on Communications*, 63(5):1676–1689, 2015.

[48] P Ubaidulla and Ananthanarayanan Chockalingam. Relay precoder optimization in MIMO-relay networks with imperfect csi. *IEEE Transactions on Signal Processing*, 59(11):5473–5484, 2011.

[49] Christos Masouros and Gan Zheng. Exploiting known interference as green signal power for downlink beamforming optimization. *IEEE Transactions on Signal processing*, 63(14):3628–3640, 2015.

[50] Ami Wiesel, Yonina C Eldar, and Shlomo Shamai. Linear precoding via conic optimization for fixed MIMO receivers. *IEEE transactions on signal processing*, 54(1):161–176, 2005.

[51] Max Costa. Writing on dirty paper (corresp.). *IEEE transactions on information theory*, 29(3):439–441, 1983.

[52] Hiroshi Harashima and Hiroshi Miyakawa. Matched-transmission technique for channels with intersymbol interference. *IEEE Transactions on Communications*, 20(4):774–780, 1972.

[53] Christian B Peel, Bertrand M Hochwald, and A Lee Swindlehurst. A vector-perturbation technique for near-capacity multiantenna multiuser communication-part I: channel inversion and regularization. *IEEE Transactions on Communications*, 53(1):195–202, 2005.

[54] Bertrand M Hochwald, Christian B Peel, and A Lee Swindlehurst. A vector-perturbation technique for near-capacity multiantenna multiuser communication-part ii: Perturbation. *IEEE Transactions on Communications*, 53(3):537–544, 2005.

[55] Juyul Lee and Nihar Jindal. Dirty paper coding vs. linear precoding for MIMO broadcast channels. In *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pages 779–783. IEEE, 2006.

[56] Johannes Maurer, Joakim Jaldén, Dominik Seethaler, and Gerald Matz. Vector perturbation precoding revisited. *IEEE Transactions on Signal Processing*, 59(1):315–328, 2010.

[57] Christos Masouros, Mathini Sellathurai, and Tharmalingam Ratnarajah. Interference optimization for transmit power reduction in Tomlinson-

Harashima Precoded MIMO downlinks. *IEEE transactions on signal processing*, 60(5):2470–2481, 2012.

[58] Christos Masouros, Mathini Sellathurai, and Tharmalingam Ratnarajah. Computationally efficient vector perturbation precoding using thresholded optimization. *IEEE transactions on communications*, 61(5):1880–1890, 2013.

[59] Adrian Garcia-Rodriguez and Christos Masouros. Power-efficient Tomlinson-Harashima precoding for the downlink of multi-user MISO systems. *IEEE transactions on communications*, 62(6):1884–1896, 2014.

[60] Christos Masouros, Mathini Sellathurai, and Tharmalingam Ratnarajah. Vector perturbation based on symbol scaling for limited feedback MISO downlinks. *IEEE Transactions on Signal Processing*, 62(3):562–571, 2014.

[61] Christos Masouros. Correlation rotation linear precoding for MIMO broadcast communications. *IEEE Transactions on Signal Processing*, 59(1):252–262, 2010.

[62] Chris TK Ng and Howard Huang. Linear precoding in cooperative MIMO cellular networks with limited coordination clusters. *IEEE Journal on Selected Areas in communications*, 28(9):1446–1454, 2010.

[63] Christos Masouros and Emad Alsusa. Dynamic linear precoding for the exploitation of known interference in MIMO broadcast systems. *IEEE Transactions on Wireless Communications*, 8(3):1396–1404, 2009.

[64] Christos Masouros, Mathini Sellathurai, and Tharmalingam Ratnarajah. A low-complexity sequential encoder for threshold vector perturbation. *IEEE communications letters*, 17(12):2225–2228, 2013.

[65] Maha Alodeh, Symeon Chatzinotas, and Björn Ottersten. Constructive multiuser interference in symbol level precoding for the MISO downlink channel. *IEEE Transactions on Signal processing*, 63(9):2239–2252, 2015.

[66] Michael Joham, Wolfgang Utschick, and Josef A Nossek. Linear transmit processing in MIMO communications systems. *IEEE Transactions on signal Processing*, 53(8):2700–2712, 2005.

[67] Gerard J Foschini, Glenn D Golden, Reinaldo A Valenzuela, and Peter W Wolniansky. Simplified processing for high spectral efficiency wireless communication employing multi-element arrays. *IEEE Journal on Selected areas in communications*, 17(11):1841–1852, 1999.

[68] Wei Yu, David P Varodayan, and John M Cioffi. Trellis and convolutional precoding for transmitter-based interference presubtraction. *IEEE Transactions on Communications*, 53(7):1220–1230, 2005.

[69] Farrokh Rashid-Farrokhi, KJ Ray Liu, and Leandros Tassiulas. Transmit beamforming and power control for cellular wireless systems. *IEEE Journal on selected areas in communications*, 16(8):1437–1450, 1998.

[70] Eugene Visotsky and Upamanyu Madhow. Optimum beamforming using transmit antenna arrays. In *1999 IEEE 49th Vehicular Technology Conference (Cat. No. 99CH36363)*, volume 1, pages 851–856. IEEE, 1999.

[71] Mats Bengtsson and Bjorn Ottersten. Handbook of antennas in wireless communications. In *Optimal and Suboptimal Transmit Beamforming*, volume 18. CRC press Boco Raton, FL, USA, 2001.

[72] Alex B Gershman, Nicholas D Sidiropoulos, Shahram Shahbazpanahi, Mats Bengtsson, and Bjorn Ottersten. Convex optimization-based beamforming. *IEEE Signal Processing Magazine*, 27(3):62–75, 2010.

[73] Emil Björnson, Mats Bengtsson, and Björn Ottersten. Optimal multiuser transmit beamforming: A difficult problem with a simple solution structure [lecture notes]. *IEEE Signal Processing Magazine*, 31(4):142–148, 2014.

[74] Antonio Pascual-Iserte, Daniel Pérez Palomar, Ana I Pérez-Neira, and Miguel Ángel Lagunas. A robust maximin approach for MIMO communica-

tions with imperfect channel state information based on convex optimization. *IEEE Transactions on Signal Processing*, 54(1):346–360, 2005.

[75] Batu K Chalise, Shahram Shahbazpanahi, Andreas Czylwik, and Alex B Gershman. Robust downlink beamforming based on outage probability specifications. *IEEE Transactions on Wireless Communications*, 6(10):3498–3503, 2007.

[76] Nikola Vucic and Holger Boche. Robust qos-constrained optimization of downlink multiuser MISO systems. *IEEE Transactions on Signal Processing*, 57(2):714–725, 2008.

[77] Gan Zheng, Kai-Kit Wong, and Tung-Sang Ng. Robust linear MIMO in the downlink: A worst-case optimization with ellipsoidal uncertainty regions. *EURASIP Journal on Advances in Signal Processing*, 2008:1–15, 2008.

[78] Ami Wiesel, Yonina C Eldar, and Shlomo Shamai Shitz. Optimization of the MIMO compound capacity. *IEEE transactions on wireless communications*, 6(3):1094–1101, 2007.

[79] Søren Skovgaard Christensen, Rajiv Agarwal, Elisabeth De Carvalho, and John M Cioffi. Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design. *IEEE Transactions on Wireless Communications*, 7(12):4792–4799, 2008.

[80] Christos Masouros and Emad Alsusa. Two-stage transmitter precoding based on data-driven code-hopping and partial zero forcing beamforming for MC-CDMA communications. *IEEE Transactions on Wireless Communications*, 8(7):3634–3645, 2009.

[81] Christos Masouros and Emad Alsusa. Soft linear precoding for the downlink of DS/CDMA communication systems. *IEEE transactions on vehicular technology*, 59(1):203–215, 2009.

[82] Ang Li and Christos Masouros. Exploiting constructive mutual coupling in P2P MIMO by analog-digital phase alignment. *IEEE Transactions on Wireless Communications*, 16(3):1948–1962, 2017.

[83] Danilo Spano, Maha Alodeh, Symeon Chatzinotas, and Björn Ottersten. Symbol-level precoding for the nonlinear multiuser MISO downlink channel. *IEEE Transactions on Signal Processing*, 66(5):1331–1345, 2017.

[84] Ang Li, Christos Masouros, Yonghui Li, and Branka Vucetic. Interference exploitation precoding for multi-level modulations. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4679–4683. IEEE, 2019.

[85] Ang Li, Danilo Spano, Jevgenij Krivochiza, Stavros Domouchtsidis, Christos G Tsinos, Christos Masouros, Symeon Chatzinotas, Yonghui Li, Branka Vucetic, and Björn Ottersten. A tutorial on interference exploitation via symbol-level precoding: overview, state-of-the-art and future directions. *IEEE Communications Surveys & Tutorials*, 22(2):796–839, 2020.

[86] Hao Huang, Wenchao Xia, Jian Xiong, Jie Yang, Gan Zheng, and Xiaomei Zhu. Unsupervised learning-based fast beamforming design for downlink MIMO. *IEEE Access*, 7:7599–7605, 2018.

[87] Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nikos D Sidiropoulos. Learning to optimize: Training deep neural networks for wireless resource management. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–6. IEEE, 2017.

[88] Abderrahmane Mayouche, Danilo Spano, Christos G Tsinos, Symeon Chatzinotas, and Bjorn Ottersten. Machine learning assisted physec attacks and SLP countermeasures for multi-antenna downlink systems. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[89] Xiao Li, Xiaoxiang Yu, Tingting Sun, Jiajia Guo, and Jun Zhang. Joint scheduling and deep learning-based beamforming for FD-MIMO systems over correlated rician fading. *IEEE Access*, 7:118297–118309, 2019.

[90] Minghe Zhu and Tsung-Hui Chang. Optimization inspired learning network for multiuser robust beamforming. In *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 1–5. IEEE, 2020.

[91] Xueyuan Wang and M Cenk Gursoy. Multi-agent double deep q-learning for beamforming in mmwave MIMO networks. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–6. IEEE, 2020.

[92] Foad Sohrabi, Hei Victor Cheng, and Wei Yu. Robust symbol-level precoding via autoencoder-based deep learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8951–8955. IEEE, 2020.

[93] Hamed Hojatian, Jérémy Nadal, Jean-François Frigon, and François Leduc-Primeau. Unsupervised deep learning for massive MIMO hybrid beamforming. *IEEE Transactions on Wireless Communications*, 2021.

[94] Junchao Shi, Wenjin Wang, Xinping Yi, Xiqi Gao, and Geoffrey Ye Li. Deep learning-based robust precoding for massive MIMO. *IEEE Transactions on Communications*, 69(11):7429–7443, 2021.

[95] Juping Zhang, Minglei You, Gan Zheng, Ioannis Krikidis, and Liqiang Zhao. Model-driven learning for generic MIMO downlink beamforming with uplink channel information. *IEEE Transactions on Wireless Communications*, 2021.

[96] Lissy Pellaco, Mats Bengtsson, and Joakim Jaldén. Deep weighted MMSE downlink beamforming. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4915–4919. IEEE, 2021.

[97] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[98] Huaiyu Dai, Andreas F Molisch, and H Vincent Poor. Downlink capacity of interference-limited MIMO systems with joint detection. *IEEE Transactions on Wireless Communications*, 3(2):442–453, 2004.

[99] Xu Zhu and Ross D Murch. Performance analysis of maximum likelihood detection in a MIMO antenna system. *IEEE Transactions on Communications*, 50(2):187–191, 2002.

[100] Fredrik Rusek, Daniel Persson, Buon Kiong Lau, Erik G Larsson, Thomas L Marzetta, Ove Edfors, and Fredrik Tufvesson. Scaling up MIMO: Opportunities and challenges with very large arrays. *IEEE signal processing magazine*, 30(1):40–60, 2012.

[101] Ricardo Tadashi Kobayashi, Fernando Ciriaco, and Taufik Abrão. Performance and complexity analysis of sub-optimum MIMO detectors under correlated channel. In *2014 International Telecommunications Symposium (ITS)*, pages 1–5. IEEE, 2014.

[102] Yeon-Geun Lim, Chan-Byoung Chae, and Giuseppe Caire. Performance analysis of massive MIMO for cell-boundary users. *IEEE Transactions on Wireless Communications*, 14(12):6827–6842, 2015.

[103] Andreas Burg, Moritz Borgmann, Markus Wenk, Martin Zellweger, Wolfgang Fichtner, and Helmut Bolcskei. Vlsi implementation of mimo detection using the sphere decoding algorithm. *IEEE Journal of solid-state circuits*, 40(7):1566–1577, 2005.

[104] Joakim Jaldén and Björn Ottersten. On the complexity of sphere decoding in digital communications. *IEEE transactions on signal processing*, 53(4):1474–1484, 2005.

[105] Ami Wiesel, Yonina C Eldar, and Shlomo Shamai. Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels. *IEEE Signal Processing Letters*, 12(9):653–656, 2005.

[106] Wing-Kin Ma, Pak-Chung Ching, and Zhi Ding. Semidefinite relaxation based multiuser detection for M-ary PSK multiuser systems. *IEEE Transactions on Signal Processing*, 52(10):2862–2872, 2004.

[107] Joakim Jaldén and Björn Ottersten. The diversity order of the semidefinite relaxation detector. *IEEE Transactions on Information Theory*, 54(4):1406–1422, 2008.

[108] Wing-Kin Ma, Chao-Cheng Su, Joakim Jaldén, Tsung-Hui Chang, and Chong-Yung Chi. The equivalence of semidefinite relaxation MIMO detectors for higher-order QAM. *IEEE Journal of Selected Topics in Signal Processing*, 3(6):1038–1052, 2009.

[109] David L Donoho, Adel Javanmard, and Andrea Montanari. Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing. *IEEE transactions on information theory*, 59(11):7434–7464, 2013.

[110] Sheng Wu, Linling Kuang, Zuyao Ni, Jianhua Lu, Defeng Huang, and Qinghua Guo. Low-complexity iterative detection for large-scale multiuser MIMO-OFDM systems using approximate message passing. *IEEE Journal of Selected Topics in Signal Processing*, 8(5):902–915, 2014.

[111] Chao Wei, Huaping Liu, Zaichen Zhang, Jian Dang, and Liang Wu. Approximate message passing-based joint user activity and data detection for NOMA. *IEEE Communications Letters*, 21(3):640–643, 2016.

[112] Zhaoyang Zhang, Xiao Cai, Chunguang Li, Caijun Zhong, and Huaiyu Dai. One-bit quantized massive MIMO detection based on variational approximate message passing. *IEEE Transactions on Signal Processing*, 66(9):2358–2373, 2017.

[113] Sundeep Rangan, Philip Schniter, Alyson K Fletcher, and Subrata Sarkar. On the convergence of approximate message passing with arbitrary matrices. *IEEE Transactions on Information Theory*, 65(9):5339–5351, 2019.

[114] Suchun Zhang, Chao-Kai Wen, Keigo Takeuchi, and Shi Jin. Orthogonal approximate message passing for GFDM detection. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2017.

[115] Mehrdad Khani, Mohammad Alizadeh, Jakob Hoydis, and Phil Fleming. Adaptive neural signal detection for massive MIMO. *IEEE Transactions on Wireless Communications*, 19(8):5635–5648, 2020.

[116] Abdullahi Mohammad, Christos Masouros, and Yiannis Andreopoulos. Complexity-scalable neural-network-based MIMO detection with learnable weight scaling. *IEEE Transactions on Communications*, 68(10):6101–6113, 2020.

[117] Neev Samuel, Tzvi Diskin, and Ami Wiesel. Learning to detect. *IEEE Transactions on Signal Processing*, 67(10):2554–2564, 2019.

[118] Timothy J. O'Shea, Tugba Erpek, and T. Charles Clancy. Deep learning based MIMO communications. *CoRR*, abs/1707.07980, 2017.

[119] Timothy O'Shea and Jakob Hoydis. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):563–575, 2017.

[120] Weihong Xu, Zhiwei Zhong, Yair Be'ery, Xiaohu You, and Chuan Zhang. Joint neural network equalizer and decoder. In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–5. IEEE, 2018.

[121] A. Balatsoukas-Stimming and C. Studer. Deep unfolding for communica-

tions systems: A survey and some new directions. In *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*, pages 266–271, 2019.

[122] Vincent Corlay, Joseph J Boutros, Philippe Ciblat, and Loc Brunel. Multilevel MIMO detection with deep learning. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 1805–1809. IEEE, 2018.

[123] Xiaosi Tan, Weihong Xu, Yair Be'ery, Zaichen Zhang, Xiaohu You, and Chuan Zhang. Improving massive MIMO belief propagation detector with deep neural network. *CoRR*, abs/1804.01002, 2018.

[124] Xiangfeng Liu and Ying Li. Deep MIMO detection based on belief propagation. In *2018 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2018.

[125] Masayuki Imanishi, Satoshi Takabe, and Tadashi Wadayama. Deep learning-aided iterative detector for massive overloaded MIMO channels. *CoRR*, abs/1806.10827, 2018.

[126] Satoshi Takabe, Masayuki Imanishi, Tadashi Wadayama, Ryo Hayakawa, and Kazunori Hayashi. Trainable projected gradient detector for massive overloaded MIMO channels: Data-driven tuning approach. *IEEE Access*, 7:93326–93338, 2019.

[127] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[128] Giancarlo Zaccone and Md Rezaul Karim. *Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python*. Packt Publishing Ltd, 2018.

[129] Sebastian Dörner, Sebastian Cammerer, Jakob Hoydis, and Stephan ten Brink. Deep learning based communication over the air. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):132–143, 2018.

[130] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[131] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[132] Li Deng and Dong Yu. Deep learning: methods and applications. *Foundations and trends in signal processing*, 7(3–4):197–387, 2014.

[133] Ephraim Zehavi. 8-psk trellis codes for a rayleigh channel. *IEEE Transactions on Communications*, 40(5):873–884, 1992.

[134] Timothy J O'Shea, Kiran Karra, and T Charles Clancy. Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention. In *2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 223–228. IEEE, 2016.

[135] Tianqi Wang, Chao-Kai Wen, Hanqing Wang, Feifei Gao, Tao Jiang, and Shi Jin. Deep learning for wireless physical layer: Opportunities and challenges. *China Communications*, 14(11):92–111, 2017.

[136] Ying He, Zheng Zhang, F Richard Yu, Nan Zhao, Hongxi Yin, Victor CM Leung, and Yanhua Zhang. Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks. *IEEE Transactions on Vehicular Technology*, 66(11):10433–10445, 2017.

[137] Gihan J Mendis, Jin Wei, and Arjuna Madanayake. Deep learning-based automated modulation classification for cognitive radio. In *2016 IEEE International Conference on Communication Systems (ICCS)*, pages 1–6. IEEE, 2016.

[138] Kevin Merchant, Shauna Revay, George Stantchev, and Bryan Nousain. Deep learning for rf device fingerprinting in cognitive communication networks. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):160–167, 2018.

[139] Sebastian Dörner, Sebastian Cammerer, Jakob Hoydis, and Stephan Ten Brink. Deep learning based communication over the air. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):132–143, 2017.

[140] Chao Shang, Fan Yang, Dexian Huang, and Wenxiang Lyu. Data-driven soft sensor development based on deep learning technique. *Journal of Process Control*, 24(3):223–233, 2014.

[141] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep ADMM-Net for compressive sensing MRI. In *Proceedings of the 30th international conference on neural information processing systems*, pages 10–18, 2016.

[142] Zongben Xu and Jian Sun. Model-driven deep-learning. *National Science Review*, 5(1):22–24, 2018.

[143] Xisuo Ma and Zhen Gao. Data-driven deep learning to design pilot and channel estimator for massive mimo. *IEEE Transactions on Vehicular Technology*, 69(5):5677–5682, 2020.

[144] Xiaowei Xu, Yukun Ding, Sharon Xiaobo Hu, Michael Niemier, Jason Cong, Yu Hu, and Yiyu Shi. Scaling for edge inference of deep neural networks. *Nature Electronics*, 1(4):216–222, 2018.

[145] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 18(1):6869–6898, 2017.

[146] Abdullahi Mohammad, Christos Masouros, and Yiannis Andreopoulos. Accelerated learning-based MIMO detection through weighted neural network design. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[147] Tahmid Abtahi, Amey Kulkarni, and Tinoosh Mohsenin. Accelerating con-

volutional neural network with FFT on tiny cores. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4. IEEE, 2017.

[148] Ehud D Karnin. A simple procedure for pruning back-propagation trained neural networks. *IEEE transactions on neural networks*, 1(2):239–242, 1990.

[149] Christos Masouros and Tharmalingam Ratnarajah. Interference as a source of green signal power in cognitive relay assisted co-existing mimo wireless transmissions. *IEEE Transactions on Communications*, 60(2):525–536, 2011.

[150] Christos Masouros. Harvesting signal power from constructive interference in multiuser downlinks. In *Wireless Information and Power Transfer: A New Paradigm for Green Communications*, pages 87–122. Springer, 2018.

[151] Yiyang Ni, Shi Jin, Wei Xu, Yuyang Wang, Michail Matthaiou, and Hongbo Zhu. Beamforming and interference cancellation for D2D communication underlaying cellular networks. *IEEE Transactions on Communications*, 64(2):832–846, 2015.

[152] Steven Hong, Joel Brand, Jung Il Choi, Mayank Jain, Jeff Mehlman, Sachin Katti, and Philip Levis. Applications of self-interference cancellation in 5G and beyond. *IEEE Communications Magazine*, 52(2):114–121, 2014.

[153] Christos Masouros and Emad Alsusa. A novel transmitter-based selective-precoding technique for DS/CDMA systems. In *2007 IEEE International Conference on Communications*, pages 2829–2834. IEEE, 2007.

[154] C. Masouros and E. Alsusa. Dynamic linear precoding for the exploitation of known interference in mimo broadcast systems. *IEEE Transactions on Wireless Communications*, 8(3):1396–1404, 2009.

[155] Ang Li and Christos Masouros. A two-stage vector perturbation scheme for adaptive modulation in downlink mu-mimo. *IEEE Transactions on Vehicular Technology*, 65(9):7785–7791, 2015.

[156] Stelios Timotheou, Gan Zheng, Christos Masouros, and Ioannis Krikidis. Exploiting constructive interference for simultaneous wireless information and power transfer in multiuser downlink systems. *IEEE Journal on Selected Areas in Communications*, 34(5):1772–1784, 2016.

[157] Ashkan Kalantari, Christos Tsinos, Mojtaba Soltanalian, Symeon Chatzinotas, Wing-Kin Ma, and Björn Ottersten. Spatial peak power minimization for relaxed phase M-PSK MIMO directional modulation transmitter. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2011–2015. IEEE, 2017.

[158] Ka Lung Law, Christos Masouros, and Marius Pesavento. Transmit precoding for interference exploitation in the underlay cognitive radio z-channel. *IEEE Transactions on Signal Processing*, 65(14):3617–3631, 2017.

[159] Carla Bertocchi, Emilie Chouzenoux, Marie-Caroline Corbineau, Jean-Christophe Pesquet, and Marco Prato. Deep unfolding of a proximal interior point method for image restoration. *Inverse Problems*, 36(3):034005, 2020.

[160] Raphael Hauser. Interior-point methods for inequality constrained optimization, 2007.

[161] Nelly Pustelnik and Laurent Condat. Proximity operator of a sum of functions; application to depth map estimation. *IEEE Signal Processing Letters*, 24(12):1827–1831, 2017.

[162] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[163] Alexander Kaplan and Rainer Tichatschke. Proximal methods in view of interior-point strategies. *Journal of optimization theory and applications*, 98(2):399–429, 1998.

[164] Sergey Ioffe and Christian Szegedy. Batch normalisation: Accelerating deep

network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[165] Hao Zheng, Zhanlei Yang, Wenju Liu, Jizhong Liang, and Yanpeng Li. Improving deep neural networks using softplus units. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–4. IEEE, 2015.

[166] Baiyang Wang and Diego Klabjan. Regularization for unsupervised deep neural nets. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2681–2681, 2017.

[167] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[168] Kun-Yu Wang, Anthony Man-Cho So, Tsung-Hui Chang, Wing-Kin Ma, and Chong-Yung Chi. Outage constrained robust transmit optimization for multiuser MISO downlinks: Tractable approximations by conic optimization. *IEEE Transactions on Signal Processing*, 62(21):5690–5705, 2014.

[169] Mark Borgerding, Philip Schniter, and Sundeep Rangan. AMP-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16):4293–4308, 2017.

[170] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

[171] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017.

[172] Yinpeng Dong, Renkun Ni, Jianguo Li, Yurong Chen, Hang Su, and Jun Zhu. Stochastic quantization for learning accurate low-bit deep neural networks. *International Journal of Computer Vision*, 127(11):1629–1642, 2019.

[173] Abdullahi Mohammad, Christos Masouros, and Yiannis Andreopoulos. *An Unsupervised Deep Unfolding Framework for Robust Symbol Level Precoding*, 2021.

[174] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: training deep neural networks with binary weights during propagations. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 3123–3131, 2015.

[175] Hande Alemdar, Vincent Leroy, Adrien Prost-Boucle, and Frédéric Pétrot. Ternary neural networks for resource-efficient AI applications. In *2017 international joint conference on neural networks (IJCNN)*, pages 2547–2554. IEEE, 2017.

[176] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[177] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in neural information processing systems*, pages 4107–4115, 2016.

[178] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. CUDA, release: 10.2.89, 2020.

[179] Joseph Bethge, Haojin Yang, Marvin Bornstein, and Christoph Meinel. Binarydensenet: developing an architecture for binary neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[180] Huan Yao and Gregory W Wornell. Lattice-reduction-aided detectors for MIMO communication systems. In *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, volume 1, pages 424–428. IEEE, 2002.

[181] Thomas Kailath, Haris Vikalo, and Babak Hassibi. MIMO receive algorithms. *Space-Time Wireless Systems: From Array Processing to MIMO Communications*, 3:2, 2005.

[182] Christoph Windpassinger, Lutz Lampe, Robert FH Fischer, and Thorsten Hehn. A performance study of MIMO detectors. *IEEE Transactions on Wireless Communications*, 5(8):2004–2008, 2006.

[183] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[184] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013.

[185] Brad McDanel, Surat Teerapittayanon, and HT Kung. Incomplete dot products for dynamic computation scaling in neural network inference. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 186–193. IEEE, 2017.

[186] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

[187] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[188] Benjamin D. Haeffele and Rene Vidal. Global optimality in neural network training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[189] Dmitry Malioutov and Aleksandr Aravkin. Iterative log thresholding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7198–7202. IEEE, 2014.

[190] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2012.

[191] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.