FACULTY OF ENGINEERING
AND ARCHITECTURE

Towards Data-Driven Communication Signal Processing in Realistic Conditions

Mostafa Naseri

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Electrical Engineering

Supervisors
Prof. Adnan Shahid, PhD - Prof. Eli De Poorter, PhD
Department of Information Technology
Faculty of Engineering and Architecture, Ghent University

April 2025

GHENT
UNIVERSITY

ISBN 978-94-6355-973-7
NUR 959
Wettelijk depot: D/2025/10.500/33

Members of the Examination Board

Chair

Prof. Sabine Wittevrongel, PhD, Ghent University

Other members entitled to vote

Prof. Rafael Berkvens, PhD, Universiteit Antwerpen
Prof. Mamoun Guenach, PhD, Ghent University
Prof. Ingrid Moerman, PhD, Ghent University
Prof. David Plets, PhD, Ghent University
Prof. Sofie Pollin, PhD, KU Leuven

Supervisors

Prof. Adnan Shahid, PhD, Ghent University
Prof. Eli De Poorter, PhD, Ghent University

Acknowledgements

Pursuing my PhD has been both demanding and profoundly rewarding—a journey I could never have navigated alone. The unwavering support, wisdom, and generosity of countless extraordinary people carried me forward.

First and foremost, my deepest gratitude goes to my family and friends. Mom and Dad, your endless patience and belief in me—even when this path felt like a perpetual maze—kept me grounded. To my sister and brothers: your rock-solid confidence in my abilities pulled me through every moment of self-doubt. To my friends: your companionship transformed these demanding years into a joyful adventure; you were my lifeline of laughter and moral support.

Professionally, this PhD research succeeded thanks to collaborations with brilliant minds at Ghent University's IDLab, Princeton University, imec, KU Leuven, the University of Málaga, and the University of Antwerp. I'm deeply grateful to my mentors for their guidance and thoughtful advice. To the reviewers and jury members: your critiques weren't always easy to absorb, but they pushed my research to heights I couldn't have reached alone.

Finally, to everyone who lent their support—whether through lively discussions, unexpected moments of clarity, or simply being there when it mattered—I extend my heartfelt thanks. You've shown me that progress isn't a solo sprint; it's a race run with shared curiosity, critique, and kindness. As I step beyond this chapter, I carry a truth shaped by all of you: the most meaningful discoveries don't belong to individuals—they're born from the sparks we ignite together.

Table of Contents

List of Figures

List of Tables

List of Acronyms

A

AI	artificial intelligence
AMC	adaptive modulation and coding
AoA	angle of arrival
AWGN	additive white Gaussian noise

B

BER	Bit Error Rate
BPP	bits per pixel

C

CCI	Co-channel Interference
CDF	cumulative distribution function
CIR	channel impulse response
CNN	convolutional neural network
CPU	central processing unit

D

DA	data augmentation
DCT	discrete cosine transform
DNN	deep neural network

F

FC fully connected
FFT fast Fourier transform
FLOPS Floating Point Operations Per Second
FoV field of view
FP first path

G

GAN generative adversarial network
GPU graphics processing unit

H

HEVC high efficiency video coding

I

IoT Internet of things
IQ In-phase and quadrature

L

LAN local area network
LIC learned image compression
LOS line-of-sight
LSTM long short-term memory

M

MACs Multiply-Accumulate Operations

MAE mean absolute error
MAGE mean absolute generalization error
MIMO Multiple Input Multiple Output
ML machine learning
MoCap motion capture
MSE Mean Squared Error

N

NLoS non-line-of-sight
NN neural network

O

OOM Out-of-Memory

P

PDoA phase difference of arrival
MUSIC multiple signal classification
PSNR peak signal-to-noise ratio

Q

QAM quadrature amplitude modulation
QAT Quantization-Aware Training
QPSK Quadrature Phase Shift Keying

R

ReLU rectified linear unit
RFIC Radio-Frequency Integrated Circuit

RNN recurrent neural network
RRC Root Raised Cosine

S

SGD stochastic gradient descent
SINR Signal-to-Interference-plus-Noise Ratio
SNR signal-to-noise ratio
SOI Signal of Interest
SSIM structural similarity index measure
SVC scalable video coding

T

TL transfer learning
ToF time-of-flight

U

UWB ultra-wideband

U

V2X Vehicle-to-Anything communication
VAE variational autoencoder
VQ vector quantization
VQGAN vector quantized generative adversarial networks
VQVAE vector quantized variational autoencoder

W

Wi-Fi IEEE 802.11
WSN Wireless Sensor Network

Samenvatting
– Summary in Dutch –

In de hedendaagse onderling verbonden wereld vormen draadloze communicatiesystemen de ruggengraat van de moderne samenleving, die alles aandrijven, van mobiele netwerken en het Internet of Things (IoT) tot kritieke systemen voor openbare veiligheid. Naarmate draadloze netwerken echter complexer worden en de vraag naar snelle en betrouwbare connectiviteit groeit, worden traditionele signaalverwerkingsmethoden steeds meer uitgedaagd door de dynamiek van multipadpropagatie, hardware-onvolkomenheden en interferentie van overlappende signalen in de praktijk. Deze dissertatie pakt deze uitdagingen aan door de kracht van geavanceerde machine learning (ML) technieken te benutten. Door deep learning-architecturen te integreren met klassieke inzichten in signaalverwerking, ontwikkelt dit werk robuuste, efficiënte en aanpasbare oplossingen op drie kerngebieden: parameterestimatie, co-channel interferentie-annulering en geleerde beeldcompressie.

Het onderzoek begint met een herbezinning op het fundamentele probleem van parameterestimatie in ultra-wideband (UWB) systemen, met een bijzondere focus op de hoek-van-aankomst (AoA) schatting. Klassieke methoden, zoals het MUSIC-algoritme en fasedifferentie-schatters, worden al lange tijd ingezet om de richting van een binnenkomend signaal te bepalen. Deze benaderingen worden echter vaak belemmerd door hardware-onvolkomenheden, polarisatiemismatches en de complexe propagatieomgevingen die in praktische toepassingen voorkomen. Als reactie hierop introduceert deze dissertatie een op een deep convolutional neural network (DCNN) gebaseerde AoA-schatter. Uitgebreide experimenten tonen aan dat de voorgestelde DCNN-schatter niet alleen de AoA-schatfout aanzienlijk vermindert — met verbeteringen tot wel 80° ten opzichte van conventionele technieken — maar ook de rekenkundige efficiëntie behoudt, met een toename in complexiteit van slechts ongeveer 10% in vergelijking met MUSIC. Deze substantiële foutvermindering benadrukt het potentieel van deep learning om de beperkingen van klassieke benaderingen te overwinnen in omgevingen waar hardware-onvolkomenheden en multipad-effecten gangbaar zijn.

Voortbouwend op het succes van de op DCNN gebaseerde AoA-schatter, pakt de dissertatie vervolgens het kritieke vraagstuk van modelgeneralizatie aan in dynamische en onvoorspelbare omgevingen. In real-world UWB localisatiesystemen weerspiegelen de trainingsomstandigheden zelden de omstandigheden die tijdens de implementatie worden aangetroffen. Variaties in kanaalkarakteristieken, multi-

padcondities en hardwareprestaties kunnen ervoor zorgen dat een model dat in één omgeving getraind is, ondermaats presteert in een andere. Om dit aan te pakken, wordt een nieuw raamwerk ontwikkeld dat data-augmentatie combineert met transfer learning-technieken. Door vooraf getrainde DCNN-modellen af te stemmen met slechts een beperkt aantal observaties uit een nieuwe omgeving, kalibreert het raamwerk de AoA-schatter effectief om tegemoet te komen aan ongeziene omstandigheden. Experimentele evaluaties tonen aan dat zelfs met slechts 5, 15 of 50 nieuwe samples de maximale generalisatiefout met respectievelijk tot 83.7%, 86.8% en 88.2% kan worden verminderd. Deze adaptieve benadering vergroot niet alleen de bruikbaarheid van ML-gebaseerde lokalisatie, maar illustreert ook de kracht van hybride pijplijnen in het behouden van prestaties te midden van variabele omgevingscondities.

De tweede belangrijke focus van deze dissertatie is het verminderen van co-channel interferentie (CCI), een fenomeen dat de signaalkwaliteit ernstig aantast wanneer meerdere transmissies plaatsvinden op hetzelfde frequentiekanaal. In moderne draadloze netwerken, waar het spectrum een schaarse hulpbron is en signalen vaak dicht opeengepakt zijn, is effectieve interferentieverwijdering cruciaal. Een van de hoofdstukken introduceert een U-Net-architectuur die specifiek is aangepast voor het scheiden van interferentiesignalen in het tijd-frequentiedomein. In deze benadering worden tijdsdomein-golfvormen omgezet in een tijd-frequentierepresentatie die de unieke kenmerken van interfererende signalen benadrukt. Door belangrijke elementen van orthogonal frequency division multiplexing (OFDM) systemen, zoals het cyclische prefix, te integreren, is de U-Net in staat om de inherente structuur van de signalen te benutten, wat resulteert in een gemiddelde verbetering van 63% in de mean squared error (MSE) ten opzichte van basismodellen over meerdere interferentietypes. Deze verbetering vertaalt zich direct naar een verbeterde demodulatie en decodering, waardoor de betrouwbaarheid van het communicatiesysteem toeneemt.

Erkennend dat interferentieverwijdering moet worden bereikt zonder onaanvaardbare rekenkosten — met name op randapparaten met beperkte middelen — ontwikkelt het onderzoek twee complementaire benaderingen. Een hoofdstuk presenteert een gestroomlijnd U-Net-model dat gebruik maakt van fast Fourier convolution blocks in combinatie met een long short-term memory (LSTM) laag in de bottleneck. Dit innovatieve ontwerp werkt in het Fourier-domein om globale kenmerken efficiënt te extraheren, waardoor de behoefte aan grote convolutiekernen of diepe netwerk lagen wordt verminderd. Als resultaat behaalt het model een verbetering van 26.5% in MSE in vergelijking met een diep U-Net-benchmark, terwijl de rekencomplexiteit aanzienlijk wordt verlaagd. Een ander hoofdstuk bouwt hierop voort door een geoptimaliseerde U-Net-architectuur voor high-throughput blinde CCI-onderdrukking voor te stellen, die specifiek is ontworpen voor implementatie op randapparaten. Dit model integreert depthwise separable convoluties, kwantisatie en pruningstrategieën om een delicate balans tussen nauwkeurigheid en efficiëntie te bereiken. Er worden twee modelvarianten ontwikkeld: een diepere, volledig convolutionele versie en een minder diep model dat wordt versterkt met een LSTM. Het LSTM-verrijkte model vertoont slechts een degradatie van 0.72% in

MSE terwijl het aantal multiply-accumulate operaties (MACs) met 58.66% wordt verminderd, en het volledig convolutionele model toont een verbetering van 0.63% in MSE met een vermindering van MACs met 61.10%. Deze innovaties zijn cruciaal om te waarborgen dat interferentieverwijdering met hoge prestaties haalbaar is op energiezuinige, middelenbeperkte platforms, waardoor de weg wordt geëffend voor real-time signaalverwerking in draadloze systemen van de volgende generatie.

Het derde deel van de dissertatie verschuift de focus naar het domein van geleerde beeldcompressie voor draadloze communicatie. In moderne netwerken is efficiënte beeldtransmissie essentieel voor een breed scala aan toepassingen, waaronder IoT-apparaten, remote monitoring en multimedia streaming. Beeldcompressie in draadloze omgevingen kent echter talrijke uitdagingen, aangezien het een balans moet vinden tussen betrouwbaarheid, doorvoersnelheid en latentie onder dynamische kanaalomstandigheden. Om deze uitdagingen aan te pakken, introduceert de dissertatie een adaptieve en progressieve pijplijn voor geleerde beeldcompressie (LIC). Er worden twee state-of-the-art modellen onderzocht: een hyperprior-gebaseerde architectuur en een Vector Quantized Generative Adversarial Network (VQGAN). Het hyperprior-model bereikt uitzonderlijke compressieprestaties door middel van verliesloze bottleneckcompressie, maar is gevoelig voor bitfouten, wat foutcorrectie of retransmissiemechanismen noodzakelijk maakt. Daarentegen vertoont de VQGAN-decoder robuuste beeldreconstructiecapaciteiten, zelfs bij afwezigheid van kanaalcodering, wat de betrouwbaarheid onder uitdagende omstandigheden verbetert. Een belangrijke bijdrage van dit werk is de ontwikkeling van progressieve versies van beide modellen, die gedeeltelijke beeldtransmissie en decodering onder suboptimale kanaalomstandigheden mogelijk maken. Deze progressieve aanpak vermindert de latentie aanzienlijk door de ontvanger direct toegang te geven tot een beeld van lagere kwaliteit, waarna latere transmissies de kwaliteit verfijnen naarmate de kanaalomstandigheden verbeteren. Experimenten, uitgevoerd met de Kodak high-resolution beelddataset onder Rayleigh fading kanaalmodellen, tonen aan dat het progressieve raamwerk niet alleen de doorvoersnelheid behoudt of verbetert, maar ook de betrouwbaarheid verhoogt en de latentie vermindert, met name in omgevingen met een lage Signal-to-Noise Ratio (SNR). Vergelijkingen met adaptive WebP illustreren dat, hoewel adaptive WebP goed presteert onder gunstige omstandigheden, het tekortschiet in barre omgevingen waar de flexibiliteit en robuustheid van de progressieve geleerde modellen duidelijk naar voren komen.

Doorheen de gehele dissertatie komt een centraal thema naar voren: de noodzaak om ML-gedreven oplossingen te ontwikkelen die niet alleen uiterst nauwkeurig zijn, maar ook robuust, efficiënt en aanpasbaar aan de onvoorspelbare aard van reële draadloze omgevingen. Elk hoofdstuk is zorgvuldig ontworpen om een of meer van de significante uitdagingen aan te pakken die de praktische implementatie van ML-modellen in draadloze communicatie belemmeren. Deze uitdagingen omvatten de schaarste en variabiliteit van data, strikte computationele beperkingen op randapparaten, de noodzaak voor modellen om te generaliseren over diverse omgevingen, en de verplichting om nieuwe technieken te integreren met bestaande systemen. Het onderzoek pakt deze uitdagingen systematisch aan door middel van innovatieve architectonische ontwerpen, geavanceerde trainingsparadigma's en

strategische optimalisatietechnieken voor modellen.

De bijdragen van deze dissertatie hebben verstrekkende implicaties voor de toekomst van draadloze communicatie. Door aan te tonen dat deep learning modellen kunnen worden aangepast om de inherente beperkingen van traditionele signaalverwerkingsmethoden te overwinnen, legt dit werk de basis voor communicatiesystemen van de volgende generatie die betrouwbaarder, efficiënter en schaalbaarder zijn. De nieuwe op DCNN gebaseerde AoA-schatters, adaptieve raamwerken voor ongeziene omgevingen en geoptimaliseerde U-Net-architecturen voor interferentieverwijdering illustreren gezamenlijk hoe ML kan worden ingezet om te voldoen aan de steeds strengere prestatie-eisen van moderne draadloze netwerken. Evenzo opent de adaptieve en progressieve beeldcompressiepijplijn die in dit werk wordt gepresenteerd nieuwe wegen om te zorgen voor hoogwaardige multimedia transmissie, zelfs onder ongunstige kanaalomstandigheden.

In wezen vertegenwoordigt deze dissertatie een belangrijke stap vooruit in de integratie van machine learning met draadloze communicatie. Het biedt een uitgebreid raamwerk dat niet alleen het theoretisch begrip vooruithelpt, maar ook praktische, inzetbare oplossingen biedt. Het in deze studie gepresenteerde onderzoek toont aan dat, door zorgvuldig de afwegingen tussen nauwkeurigheid, rekencomplexiteit en robuustheid te balanceren, het mogelijk is om ML-gedreven systemen te ontwerpen die voldoen aan de eisen van draadloze toepassingen van de volgende generatie. Naarmate draadloze netwerken blijven evolueren en uitbreiden, zullen de inzichten en methodologieën die in dit werk zijn ontwikkeld dienen als een waardevolle gids voor toekomstig onderzoek en innovatie in dit veld.

Summary

In today's interconnected world, wireless communications form the backbone of modern society, powering everything from mobile networks and the Internet of Things (IoT) to mission-critical public safety systems. Yet, as wireless networks become more complex and the demand for rapid, reliable connectivity grows, traditional signal processing methods are increasingly challenged by the real-world dynamics of multipath propagation, hardware imperfections, and interference from overlapping signals. This dissertation addresses these challenges by harnessing the power of advanced machine learning (ML) techniques. By integrating deep learning architectures with classical signal processing insights, this work develops robust, efficient, and adaptable solutions in three key areas: parameter estimation, co-channel interference cancellation, and learned image compression.

The research begins by reexamining the fundamental problem of parameter estimation in ultra-wideband (UWB) systems, with a particular focus on angle-of-arrival (AoA) estimation. Classical methods such as the MUSIC algorithm and phase difference estimators have long been relied upon to determine the direction from which a signal is received. However, these approaches are often hampered by hardware non-idealities, polarization mismatches, and the complex propagation environments encountered in practical applications. In response, this dissertation introduces a deep convolutional neural network (DCNN) based AoA estimator. Extensive experiments demonstrate that the proposed DCNN estimator not only significantly reduces the AoA estimation error—achieving improvements of up to 80° over conventional techniques—but also maintains computational efficiency, increasing complexity by only approximately 10% compared to MUSIC. This substantial reduction in error underscores the potential of deep learning to surmount the limitations of classical approaches in environments where radio frequency hardware imperfections and multipath effects are prevalent.

Building on the success of the DCNN-based AoA estimator, the dissertation then confronts the critical issue of model generalization in dynamic and unpredictable environments. In real-world UWB localization systems, the training conditions rarely mirror those encountered during deployment. Variations in channel characteristics, multipath conditions, and hardware performance can cause a model trained in one environment to underperform in another. To address this, a novel framework is developed that combines data augmentation with transfer learning techniques. By fine-tuning pre-trained DCNN models using only a limited number of observations from a new environment, the framework effectively recalibrates the AoA estimator to accommodate unseen conditions. Experimental evaluations reveal

that even with as few as 5, 15, or 50 new samples, the maximum generalization error can be reduced by up to 83.7%, 86.8%, and 88.2%, respectively. This adaptive approach not only extends the utility of ML-based localization but also illustrates the power of hybrid pipelines in maintaining performance amidst environmental variability.

The dissertation's second major focus is on mitigating co-channel interference (CCI), a phenomenon that severely degrades signal quality when multiple transmissions occur on the same frequency channel. In modern wireless networks, where spectrum is a scarce resource and signals are often closely packed, effective interference cancellation is crucial. One of the chapters introduces a U-Net architecture specifically tailored for time-frequency interference signal separation. In this approach, time-domain waveforms are converted into a time-frequency representation that highlights the unique signatures of interfering signals. By incorporating key elements of orthogonal frequency division multiplexing (OFDM) systems, such as the cyclic prefix, the U-Net is able to exploit the inherent structure of the signals, resulting in an average 63% improvement in mean squared error (MSE) performance over baseline models across multiple interference types. This improvement directly translates into enhanced demodulation and decoding performance, thereby increasing the reliability of the communication system.

Recognizing that interference cancellation must be achieved without incurring prohibitive computational costs, especially on resource-constrained edge devices, the research further develops two complementary approaches. One chapter presents a streamlined U-Net model that leverages fast Fourier convolution blocks combined with a long short-term memory (LSTM) layer in the bottleneck. This innovative design operates in the Fourier domain to extract global features efficiently, thereby reducing the need for large convolutional kernels or deep network layers. As a result, the model achieves a 26.5% improvement in MSE compared to a deep U-Net benchmark while significantly lowering computational complexity. Another chapter builds upon this by proposing an optimized U-Net architecture for high-throughput blind CCI cancellation that is specifically designed for deployment on edge devices. This model incorporates depthwise separable convolutions, quantization, and pruning strategies to achieve a delicate balance between accuracy and efficiency. Two model variants are developed: a deeper, fully convolutional version and a shallower model augmented with an LSTM. The LSTM-enhanced model exhibits only a 0.72% degradation in MSE while reducing the number of multiply-accumulate operations (MACs) by 58.66%, and the fully convolutional model shows a 0.63% improvement in MSE with a 61.10% reduction in MACs. These innovations are crucial for ensuring that high-performance interference cancellation is feasible on low-power, resource-limited platforms, thereby paving the way for real-time signal processing in next-generation wireless systems.

The third part of the dissertation shifts focus to the domain of learned image compression for wireless communications. In modern networks, efficient image transmission is essential for a wide range of applications, including IoT devices, remote monitoring, and multimedia streaming. However, image compression in wireless environments is fraught with challenges, as it must balance

reliability, throughput, and latency under dynamic channel conditions. To address these challenges, the dissertation introduces an adaptive and progressive pipeline for learned image compression (LIC). Two state-of-the-art models are examined: a hyperprior-based architecture and a Vector Quantized Generative Adversarial Network (VQGAN). The hyperprior model achieves exceptional compression performance through lossless bottleneck compression but is susceptible to bit errors, which necessitates error correction or retransmission mechanisms. In contrast, the VQGAN decoder exhibits robust image reconstruction capabilities even in the absence of channel coding, thereby enhancing reliability under challenging conditions. A key contribution of this work is the development of progressive versions of both models, which enable partial image transmission and decoding under suboptimal channel conditions. This progressive approach significantly reduces latency by allowing the receiver to access a low-fidelity image immediately, with subsequent transmissions refining the quality as channel conditions improve. Experiments conducted using the Kodak high-resolution image dataset under Rayleigh fading channel models demonstrate that the progressive framework not only maintains or improves throughput but also enhances reliability and reduces latency, particularly in low Signal-to-Noise Ratio (SNR) environments. Comparisons with adaptive WebP illustrate that while adaptive WebP performs well under favorable conditions, it falls short in harsh environments where the flexibility and robustness of the progressive learned models become evident.

Throughout the dissertation, a central theme emerges: the need to develop ML-driven solutions that are not only highly accurate but also robust, efficient, and adaptable to the unpredictable nature of real-world wireless environments. Each chapter is carefully designed to address one or more of the significant challenges that hinder the practical deployment of ML models in wireless communications. These challenges include the scarcity and variability of data, stringent computational constraints on edge devices, the need for models to generalize across diverse environments, and the imperative to integrate new techniques with legacy systems. The research systematically addresses these challenges through innovative architectural designs, advanced training paradigms, and strategic model optimization techniques.

By demonstrating that deep learning models can be adapted to overcome the inherent limitations of traditional signal processing methods, this work lays the foundation for next-generation communication systems that are more reliable, efficient, and scalable. The novel DCNN-based AoA estimators, adaptive frameworks for unseen environments, and optimized U-Net architectures for interference cancellation collectively illustrate how ML can be leveraged to meet the increasingly stringent performance requirements of modern wireless networks. Similarly, the adaptive and progressive image compression pipeline presented in this work opens up new avenues for ensuring high-quality multimedia transmission even under adverse channel conditions.

In essence, this dissertation represents a significant step forward in the integration of machine learning with wireless communications. It offers a comprehensive framework that not only advances theoretical understanding but also provides prac-

tical, deployable solutions. The research presented herein demonstrates that by carefully balancing the trade-offs between accuracy, computational complexity, and robustness, it is possible to design ML-driven systems that meet the demands of next-generation wireless applications. As wireless networks continue to evolve and expand, the insights and methodologies developed in this work will serve as a valuable guide for future research and innovation in the field.

1 Introduction

This chapter provides a broad overview of the themes and motivations behind this dissertation. We begin by examining the role of Machine Learning (ML) in modern engineering, discussing how data-driven methods have grown to complement or even surpass traditional model-based approaches. Subsequently, we highlight how these approaches find natural applications in wireless communications, bridging classical signal processing techniques with powerful deep learning architectures. By doing so, we set the stage for the more specialized discussions in subsequent chapters, where we delve into parameter estimation, interference cancellation, and learned image compression.

1.1 Overview of Machine Learning in Engineering

Machine Learning has become a cornerstone in diverse engineering fields, transforming how we approach design, optimization, and real-time decision-making [1]. At its core, ML seeks to learn patterns and make predictions or decisions based on data, rather than relying solely on explicit analytical models. This paradigm shift has been fueled by multiple factors: the availability of large datasets, advances in high-performance computing (e.g., Graphics Processing Units, specialized AI hardware), and theoretical breakthroughs in neural networks [2].

Broadly, ML approaches can be classified into three main categories [3]. Supervised learning focuses on learning a function that maps inputs $\mathbf{x}$ to desired outputs $\mathbf{y}$ using labeled examples. A common objective is to minimize a loss function

$L(\hat{\mathbf{y}}, \mathbf{y})$, such as mean squared error or cross-entropy. Unsupervised learning aims to discover intrinsic patterns in unlabeled data (e.g., clustering, dimensionality reduction). Here, there are no explicit labels, and the goal is to learn representations or structures directly from the input. Reinforcement learning trains an agent to interact with an environment by taking actions to maximize a cumulative reward, typically through trial and error.

A special class of ML methods that has garnered widespread attention across these categories is *deep learning*. Deep learning employs neural networks with multiple hidden layers, leveraging large-scale data to learn complex, hierarchical representations. A deep neural network (DNN) can be viewed as a function:

$$f(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{y}, \tag{1.1}$$

where $\mathbf{x}$ is the input (e.g., a signal, an image, or a feature vector), $\mathbf{y}$ is the desired output, and $\boldsymbol{\theta}$ represents the tunable parameters (weights and biases). Through multiple stacked layers of linear transformations and nonlinear activations, a DNN can approximate highly intricate functions. The training process adjusts $\boldsymbol{\theta}$ to minimize a loss function, commonly via gradient descent-based algorithms:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} L\big(f(\mathbf{x}_i; \boldsymbol{\theta}), \mathbf{y}_i\big), \tag{1.2}$$

where $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ is the training dataset.

This learned approach offers several advantages over classical engineering methods. Deep learning models offer several key advantages. For instance, Adaptability allows a trained deep model to be fine-tuned with relative ease to handle new data or adjust to changing environments—often a simpler task than rewriting traditional analytic models. Moreover, deep networks demonstrate significant expressive power by capturing highly nonlinear relationships that standard parametric functions may struggle to represent. In addition, they promote Automation by learning feature extraction automatically through hierarchical layer representations, thereby reducing the need for extensive domain expertise.

Despite its promise, ML often encounters serious obstacles in engineering contexts among them is the difficulty of obtaining large, high-quality datasets, which can be prohibitively expensive or logistically infeasible. Additionally, issues related to Generalization and Robustness arise since models trained in controlled settings may fail to generalize under real-world distribution shifts. There are also Resource Constraints to consider, given that many engineering applications demand low latency and limited memory usage, which poses challenges for large models. Finally, the concerns of Interpretability and Reliability are paramount, as engineering tasks often require a clear understanding of the model's decision process along with predictable performance bounds.

These challenges underscore the importance of developing ML methods that are both powerful and efficient, ensuring viability in resource-constrained or dynamically changing environments. In the following sections, we explore how these principles and challenges manifest in signal processing and wireless communication tasks, where robust parameter estimation, interference management, and data compression are of paramount importance.

1.1.1 Historical Context and Motivation

Machine Learning has roots tracing back to early attempts at statistical pattern recognition and adaptive algorithms [1]. Initial milestones included the use of linear regression and perceptrons, concepts that laid the groundwork for modern neural networks. However, for many years, the adoption of ML in engineering practice was hampered by limited computational power and the scarcity of large, high-quality datasets.

With the advent of affordable high-performance computing, propelled by *Moore's law* and the ubiquity of Graphics Processing Units (GPUs), the scene was set for rapid progress. Researchers and industry practitioners were able to train deep models on orders of magnitude more parameters and on vastly larger datasets than ever before. This period gave rise to the so-called "deep learning revolution," leading to breakthroughs in fields such as computer vision, natural language processing, and, more recently, wireless communications and signal processing.

In parallel, engineering disciplines have become increasingly data-centric. High-fidelity sensors, automated data acquisition, and internet-scale data collection have produced massive datasets, inviting data-driven techniques. As a result, many long-standing problems in engineering—such as fault diagnosis in complex systems, advanced robotics control, and adaptive communication protocols—are now often tackled through ML paradigms.

These developments motivated a fundamental reevaluation of traditional approaches in signal processing, networking, and communications. Instead of purely analytical models based on simplifying assumptions, data-driven solutions now promise to handle complex, real-world phenomena with greater flexibility and accuracy. Nevertheless, these same methods introduce new questions around robustness, interpretability, and computational feasibility—particularly in domains like wireless communications, where resources (e.g., power, bandwidth, and hardware) are often constrained. As we shall see, striking an optimal balance between model complexity, performance, and resource usage remains a central challenge [4].

1.1.2 Deep Learning Fundamentals

Deep Learning is a subfield of machine learning characterized by neural networks with multiple layers of nonlinear transformations [5]. Formally, a deep neural

network (DNN) can be represented as eq. 1.1 By stacking layers of linear transformations and nonlinear activations, a DNN can approximate highly complex mappings. The training process adjusts θ to minimize a loss function as explained in eq. 1.2.

Architectural Variants: From CNNs and RNNs to Transformers. Deep neural networks come in various forms, each suited to different data modalities and tasks. For instance, convolutional neural networks (CNNs) exploit weight-sharing via convolutional filters to capture spatial or temporal correlations effectively. They have been highly successful in tasks such as image classification, object detection, and segmentation, largely due to their hierarchical feature extraction and relatively efficient computation on grid-like data (e.g., images). In contrast, Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), introduce recurrent connections to process sequential data (e.g., time-series, speech) [6, 7]. By iterating over time steps and updating a hidden state, RNNs model temporal dependencies; however, they often face challenges like vanishing or exploding gradients when dealing with very long sequences. More recently, *transformers* have gained prominence by employing *multi-head self-attention* instead of traditional recurrent or convolutional operations [8]. Initially popularized in natural language processing, transformers have since been adapted to a variety of tasks including computer vision (as in Vision Transformers) and speech recognition, with their parallelizable design and ability to capture both local and global dependencies making them highly scalable to large datasets.

Transformer Training Paradigm. While training CNNs and RNNs typically involves a straightforward forward pass followed by backpropagation, *transformers* introduce additional mechanisms that require careful handling. Self-attention layers compute dependencies among all tokens in a sequence by generating *queries*, *keys*, and *values* for each token. Although this all-pairs interaction can be memory-intensive, it enables efficient parallelization compared to the sequential recursion found in RNNs. Moreover, since transformers do not rely on recurrence or convolution, they require an explicit encoding to capture sequence order—sine-cosine positional embeddings being a common example. For tasks such as language modeling, masking strategies are applied where certain tokens are masked to prevent the model from "seeing" future positions, thus enabling causal or bidirectional training objectives. Additionally, transformers are often trained at scale using large batches on distributed clusters; while this can accelerate convergence, it necessitates strategies like *gradient accumulation* or carefully tuned learning rate schedules to maintain model stability. These aspects collectively differentiate transformer training from that of CNNs or RNNs, offering parallelization benefits at the expense

of potentially higher memory usage.

The Emergence of Self-Supervised Learning. Alongside these architectural advancements, *self-supervised learning* has become an influential paradigm. Instead of relying on labeled data, models learn generalized representations from unlabeled corpora by solving proxy tasks (e.g., masked token prediction). This approach reduces the need for extensive annotation and often enhances model robustness. Although initiated in language tasks (e.g., BERT, GPT), self-supervision has been increasingly adapted to vision, speech, and signal processing domains—particularly appealing in scenarios where collecting labeled data is challenging or costly [9].

Training and Optimization of Deep Networks. Central to any deep architecture is the *training* process, typically using backpropagation. Stochastic Gradient Descent (SGD) and its variants (e.g., Adam, RMSProp) update model weights in small batches:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \big(\mathcal{L}(\boldsymbol{\theta}) \big), \tag{1.3}$$

where η is the learning rate, and $\mathcal{L}(\boldsymbol{\theta})$ is the loss over the current minibatch. Techniques such as L_2 regularization, dropout, and batch normalization reduce overfitting and improve generalization. In wireless communication contexts, domain-specific constraints (e.g., frequency selectivity) can inform custom layers or loss functions.

Deployment and Efficiency: Model Compression and Edge Inference. Despite their accuracy, modern deep networks—particularly large transformers—are often computationally heavy, posing significant challenges for real-time or resource-constrained environments such as edge devices or mobile platforms [10]. The growing interest in *large language models (LLMs) on the edge*, where entire transformer-based architectures must run on constrained hardware [11], further highlights the urgency of developing more efficient solutions. Several strategies have been developed to address this need. One approach is pruning, which involves removing less important weights or neurons to reduce the model size while often maintaining accuracy. Another strategy is quantization, where floating-point parameters are converted to lower-precision representations (e.g., INT8), thereby speeding up inference and reducing memory bandwidth requirements. Additionally, efficient layer designs—such as employing depthwise separable convolutions in CNNs or factorized attention mechanisms in transformers—serve to lower the number of required multiply-accumulate (MAC) operations. Lastly, knowledge distillation trains a smaller "student" model to replicate the outputs of a larger "teacher" model, thereby retaining much of the performance benefits of the original model while significantly lowering computational costs.

These compression and acceleration techniques are increasingly crucial for *on-device* intelligence, where latency, power consumption, and bandwidth are at a premium. By making deep networks more compact, developers can deploy advanced signal processing or natural language tasks directly on small embedded platforms. As we will see in subsequent chapters, such optimizations are also pivotal in wireless communication scenarios—be it for interference cancellation or learned compression—where real-time constraints and energy budgets make uncompressed ML models impractical.

Overall, from the foundational designs of CNNs and RNNs to the recent success of transformer-based and self-supervised methods, deep learning continues to evolve toward more flexible, scalable, and efficient paradigms. In the upcoming sections, we will illustrate how these principles translate to robust, high-performance solutions in signal processing and wireless communications, emphasizing techniques that balance accuracy with computational feasibility.

1.2 ML for Signal Processing: Opportunities and Challenges

Historically, signal processing methodologies have relied on analytical models rooted in principles such as linearity, stationarity, or specific forms of noise statistics. For instance, tasks like filtering, denoising, and spectral analysis often hinge on well-established approaches (e.g., Wiener filters, Fourier or wavelet transforms). While these techniques are powerful and theoretically grounded, they may struggle with complex or nonlinear interactions commonly found in modern communication systems and real-world sensor data.

Machine Learning (ML) offers several advantages to signal processing. It provides adaptability to nonlinear phenomena since ML models can approximate intricate mappings without requiring explicit assumptions about underlying signal statistics. Moreover, ML-based approaches enable feature learning by extracting relevant features directly from data rather than relying on hand-crafted features such as specific domain transformations or filtering procedures. Scalability and automation are also enhanced because, when large volumes of data are available, deep models can exploit subtle patterns that might remain hidden to classical designs, thereby improving accuracy or efficiency.

At the same time, several challenges arise when applying ML to signal processing. Obtaining large, high-quality labeled datasets is often difficult, and real-world signals may be contaminated by non-Gaussian noise or interference that is not well-represented in the training distribution. Even when sufficient data exists for one setting, models often fail to generalize when deployed in a different, unseen environment—a phenomenon sometimes referred to as “domain shift” that can

severely degrade performance. Additionally, computational constraints pose a significant issue since signal processing tasks frequently operate in real time or near-real time (for example, in communication systems), and large deep networks may be prohibitive in terms of inference speed or memory usage. This necessitates the use of specialized hardware accelerators or model compression techniques. Another important consideration is interpretability; while classical methods typically provide clear interpretability (as seen, for instance, in the frequency response of a filter), ML-driven models risk becoming "black boxes," complicating debugging and raising trust issues in safety-critical applications.

A promising direction to overcome these obstacles is the development of hybrid approaches that combine the strengths of analytical signal processing techniques with data-driven learning. For example, model-based network design involves unrolling iterative algorithms—such as those used in sparse coding or compressed sensing—into neural network layers, thus retaining interpretability while benefiting from learnable parameters. Similarly, physics-informed ML incorporates physical laws or known channel models into the loss function or network architecture to guide the learning process in a principled manner.

This blending of classical and modern approaches is particularly promising for next-generation wireless systems, where the ability to adapt in the face of unpredictable conditions (e.g., rapidly changing channels, interference, or device heterogeneity) is essential. As will be discussed in later chapters, ML-based solutions for signal processing are making strides in tasks like parameter estimation, interference cancellation, and efficient data (or image) compression, but carefully addressing the aforementioned challenges remains a key component of successful deployment.

1.2.1 Role of ML in Modern Signal Processing

In modern engineering systems, signal processing tasks often involve high-dimensional data, complex channel or noise models, and rapidly changing operational conditions. Traditional model-based frameworks—such as linear filters, Kalman filters, or parametric spectral estimation methods—can face limitations when the underlying assumptions (e.g., stationarity or Gaussianity) fail to hold.

Machine Learning (ML) has begun to fill existing gaps by providing flexible architectures that are capable of modeling nonlinearity and high-dimensional correlations. In many cases, ML methods are not intended to replace classical approaches entirely but rather to serve as complementary tools. For example, deep networks can be incorporated into established workflows by enhancing conventional algorithms in both pre- and post-processing stages, thereby refining outputs for tasks such as denoising or parameter estimation. In addition, hybrid pipelines can be formed in which analytical domain knowledge guides the design of network

layers—enforcing constraints like sparsity or known transformation structures—to blend interpretability with data-driven adaptation.

A recurring challenge in signal processing is addressing scenarios that deviate significantly from ideal assumptions. Channels may be time-varying, interference sources might be non-stationary, or hardware imperfections can distort signals. In these non-ideal environments, traditional closed-form solutions often become cumbersome or suboptimal, while ML excels by learning directly from empirical data and capturing nuances that hand-crafted models might overlook. Moreover, many signal processing applications require low-latency operation; for instance, wireless communication systems must estimate parameters or cancel interference within microseconds or milliseconds. Although large deep networks have achieved success in offline benchmarks, meeting real-time constraints imposes strict performance requirements on both software—through efficient model architectures and optimized inference engines—and hardware—via dedicated AI accelerators and low-power embedded devices. Consequently, techniques such as model compression and the development of efficient architectures (e.g., depthwise separable convolutions or pruned networks) have become indispensable to ensure the practicality of ML-based signal processing.

Furthermore, signal processing tasks are often mission-critical, as evidenced in domains like avionics, medical imaging, or emergency communications, where high reliability and predictable failure modes are paramount. Evaluating ML-based solutions in these contexts requires not only the assessment of mean performance metrics (such as Mean Squared Error, MSE) but also an examination of worst-case scenarios and robustness under adversarial conditions or out-of-distribution inputs. This focus on reliability has spurred research into uncertainty estimation, adversarial defenses, and domain adaptation, all aimed at ensuring that ML-driven solutions maintain strong performance even when faced with unexpected system variations. Overall, ML's role in modern signal processing extends beyond merely delivering incremental performance gains; it establishes a framework for addressing new classes of problems that are either infeasible or prohibitively complex for traditional analytic approaches, thereby paving the way for more adaptive, efficient, and intelligent signal processing pipelines. In the following section, the focus will shift to the specific challenges and solutions that emerge when ML methods intersect with wireless communication systems.

Despite the promise of Machine Learning (ML) for signal processing, deploying ML-driven solutions in practical systems often presents non-trivial hurdles. Data quality and availability remain major concerns since real-world data may be incomplete, noisy, or highly imbalanced, and labeling signals for supervised training can be prohibitively expensive, particularly when domain expertise is required or privacy concerns limit data sharing. Consequently, models trained on idealized or synthetic datasets may not perform reliably once deployed. Moreover, domain

shifts pose a critical challenge as models trained on a specific wireless channel profile or interference scenario can degrade significantly under new conditions, such as different frequency bands, non-stationary noise sources, or hardware variations; techniques like transfer learning, data augmentation, and domain adaptation have been explored to mitigate this issue. In addition, many embedded or edge devices face computational and memory constraints that hinder the efficient operation of large ML models, especially in tasks requiring real-time performance such as interference cancellation or parameter estimation. Methods including quantization, pruning, and efficient architectural designs (e.g., depthwise separable convolutions) are employed to reduce model size and inference time without substantially compromising performance. Interpretability and safety also emerge as significant challenges, as traditional model-based approaches in signal processing offer well-defined theoretical guarantees that facilitate troubleshooting and safety certification, whereas ML models often function as complex "black boxes," making it difficult to understand decision-making processes. This lack of transparency can hinder adoption in safety-critical or regulated industries such as medical, aerospace, and defense, even though recent advances in interpretable ML and explainable AI provide potential avenues for improvement. Additionally, reliability and robustness must be ensured because ML systems can be sensitive to adversarial perturbations or unexpected edge cases; for instance, in wireless communications, a clever jammer might exploit vulnerabilities by introducing signals designed to mislead the model. Addressing these vulnerabilities requires a careful combination of security analysis, defensive techniques (e.g., adversarial training), and redundancy in system design. In light of these challenges, implementing ML-driven solutions in the real world demands more than strong performance on benchmark datasets—engineers must consider adaptability, efficiency, interpretability, and safety to ensure that these models can reliably manage the variability and constraints inherent in actual deployment scenarios. Subsequent sections illustrate how these considerations specifically apply to wireless communication, a domain characterized by constantly changing channel conditions, strict latency requirements, and limited hardware resources.

1.3 Wireless Communication: A Key Application Domain

Wireless communication stands at the forefront of modern connectivity, enabling diverse applications ranging from personal mobile devices to massive machine-type communications in the Internet of Things (IoT). In many of these use cases, strict constraints on latency, energy consumption, and spectral efficiency coexist with the challenges of complex and time-varying channels, so designing robust, high-

performance wireless systems necessitates continuous innovation in both hardware and signal processing algorithms [12]. Contemporary wireless networks exhibit greater heterogeneity than ever before, as devices are required to operate across multiple frequency bands and standards. For instance, they must function over various spectra—such as sub-6 GHz and millimeter wave—and support different protocols including 4G, 5G, and Wi-Fi. In addition, high mobility scenarios like vehicular communication introduce rapid channel variations that demand real-time parameter estimation and adaptation, while the massive connectivity of IoT networks, with potentially billions of devices, exacerbates interference and resource allocation challenges. These factors contribute to non-stationary, high-dimensional data that can be difficult to model analytically.

Classical wireless communication techniques, such as linear receivers, Maximum Likelihood (ML) detection, and channel estimation based on pilot signals, have established a strong theoretical foundation. However, these traditional methods often rely on simplifying assumptions—models like independent and identically distributed (i.i.d.) Rayleigh fading, for example, may not hold true in certain real-world multipath environments. They also tend to suffer in nonlinear or high-interference scenarios, where nonlinear hardware (such as power amplifiers) or co-channel interference can degrade performance if not properly addressed, and they struggle with real-time adaptation because closed-form solutions or optimization routines may be too slow to respond to rapidly changing conditions.

To overcome these limitations, Machine Learning (ML) techniques are increasingly being deployed. Data-driven channel estimation allows neural networks to learn channel models directly from raw data, potentially outperforming traditional estimators in complex or non-Gaussian environments. Reinforcement learning is employed to optimize resource allocation and scheduling, fine-tuning strategies for power and bandwidth in multi-user networks under dynamic traffic demands. Deep networks can also identify and cancel co-channel interference, even when traditional assumptions about linearity or Gaussian noise no longer apply. Moreover, learned signal compression and coding methods—employing techniques such as variational autoencoders and adversarial methods—offer more efficient and robust representations compared to hand-engineered codecs.

The synergy with edge computing further accentuates both the promise and the challenges of integrating ML into wireless systems. In this paradigm, data processing occurs near the network's periphery on devices like IoT sensors and mobile phones, which often have limited computational resources and battery life. Consequently, models must be compact and efficient in both training and inference while still achieving real-time performance. As wireless standards progress toward higher frequencies and more intricate network topologies, the importance of adaptive, data-driven solutions continues to grow. Research directions such as *transfer learning*, *domain adaptation*, and *model compression* are crucial for ensur-

ing that ML-driven wireless systems remain robust against variability in hardware, environment, and user demand.

In the following sections, key interrelated challenges in wireless communications are highlighted, and the integration of ML to address them is illustrated. Three main problem areas—*parameter estimation*, *interference cancellation*, and *learned data compression*—are discussed in depth throughout this dissertation, demonstrating how tailored deep learning architectures can meet the stringent performance requirements of real-world wireless scenarios.

1.3.1 Wireless Systems and ML

The integration of Machine Learning (ML) into wireless systems is fueled by the potential to handle complex radio environments more efficiently than traditional methods. Whereas classical signal processing often relies on analytic models (e.g., linear approximations, Gaussian assumptions), ML-based approaches learn from empirical data. This data-driven framework enables them to adapt to real-world conditions, including nonlinear hardware responses, time-varying multipath channels, and non-stationary interference sources [13].

Several core operations in wireless systems are naturally amenable to ML. For instance, in channel estimation and equalization, neural networks can learn to compensate for complex channel distortions from raw signals rather than relying purely on pilot-based estimation, potentially improving accuracy in high-mobility or strongly frequency-selective channels. Similarly, reinforcement learning or supervised ML can dynamically allocate frequencies, adjust power levels, or schedule transmissions to optimize network throughput and reduce congestion in resource allocation and control. Deep architectures have also shown promise in interference management by effectively separating co-channel or overlapping signals through learned filtering or blind source separation, thereby surpassing traditional linear or narrowband interference models. Moreover, ML-based approaches applied to coding and compression—through learned encoders, decoders, and compression schemes—can outperform traditional codecs when processing high-dimensional signals or images, especially under adverse channel conditions.

1.3.2 Convergence of ML and Wireless

The escalating demands of wireless communication—higher data rates, lower latencies, and massive device connectivity—coincide with the growing sophistication of machine learning techniques, a convergence that is reshaping the design and operation of wireless networks by enabling adaptive and efficient solutions that were once infeasible. Modern wireless infrastructure increasingly adopts *software-defined* components, with baseband processing and radio functions virtualized on general-purpose hardware; this transition lowers the barriers for integrating

ML pipelines directly into communication systems, as seen in applications such as *deep packet inspection* and *policy-based routing*, where ML models monitor real-time network statistics and user demands. As networks expand to support the Internet of Things (IoT) and edge devices, *edge intelligence* emerges as a key paradigm, allowing ML models to run locally on edge devices or in edge servers rather than sending raw sensor data to the cloud, thereby reducing latency and conserving bandwidth. In this context, techniques like *federated learning* enable distributed nodes to collaboratively train collective models without sharing raw data, enhancing privacy and mitigating communication overhead. ML-driven tools further contribute by analyzing network states—such as traffic patterns, user locations, and interference levels—and making real-time decisions about resource allocation, which can include adjusting beamforming strategies, scheduling transmissions, or managing handoffs in cellular networks; approaches like Reinforcement Learning (RL) and multi-armed bandit strategies have proven especially popular for facilitating autonomous policy learning in complex, high-dimensional action spaces. Despite these advancements, the synergy between ML and wireless also introduces challenges. Wireless conditions can fluctuate on millisecond timescales, challenging static or infrequently updated ML models, while future networks are expected to span diverse hardware platforms—from micro-sensors to massive base stations—and support heterogeneous service requirements, such as ultrareliable low-latency communications and enhanced mobile broadband. Moreover, widespread adoption of ML-driven components will require standardization, interoperability, and potentially new regulatory frameworks to ensure safety and fairness.

Despite these hurdles, the trend is clear: wireless systems are increasingly designed and optimized using ML principles. The following sections and chapters delve deeper into specific use cases—*parameter estimation*, *interference cancellation*, and *learned image compression*—demonstrating how tailored deep learning approaches can overcome traditional limitations and meet the stringent requirements of next-generation wireless networks.

1.4 Dissertation Motivation and Linking Themes

Traditional wireless systems often rely on analytically derived methods that assume idealized conditions—stationarity, linearity, or Gaussian noise—yet practical deployments rarely conform to these assumptions. Meanwhile, machine learning (ML) techniques excel at capturing complex, nonlinear phenomena and can adapt more readily to varying conditions.

Bridging ML techniques with classical engineering wisdom lies at the heart of this work. Conventional solutions—like multiple signal classification (MUSIC) or phase-difference methods for angle of arrival (AoA) estimation—offer elegant theoretical underpinnings, but can struggle with hardware imperfections or unex-

pected interference. Conversely, purely data-driven approaches might overfit to a specific environment or prove too computationally heavy for real-time scenarios. This dissertation aims to develop *balanced* solutions that leverage the strengths of both paradigms: Achieving high performance on tasks such as parameter estimation, interference cancellation, and image compression—even under varying environmental conditions—is central to the focus on accuracy and adaptation. Equally important is ensuring that models are efficient and scalable, with the ability to be compressed, quantized, or pruned for operation on resource-constrained hardware with low latency, while robustness and generalization are prioritized to mitigate performance degradation when moving from controlled lab conditions to diverse, real-world deployments.

Although the specific applications explored in this dissertation—parameter estimation, co-channel interference cancellation, and learned image compression—may appear distinct, they are interconnected through several common themes. One key aspect is the data-driven modeling of complex signals; whether estimating angles in multipath environments or separating co-channel waveforms, deep learning architectures are employed to uncover latent signal structures. Handling environmental mismatch is another shared challenge, as out-of-distribution data and domain shifts are pervasive in wireless settings, prompting the need for transfer learning, data augmentation, or other adaptation strategies. Furthermore, computational constraints in achieving real-time performance on embedded devices necessitate the use of model compression techniques (such as pruning and quantization) and efficient neural architectures (for example, depthwise separable convolutions). Additionally, the concept of progressive or incremental processing is illustrated in scenarios like interference cancellation and image compression, where partial or progressive decoding can balance performance and latency to offer flexible trade-offs under dynamic channel conditions.

These themes underscore the *unifying vision* of integrating ML-driven solutions into practical wireless domains. Each challenge—be it AoA estimation in unseen environments or blind co-channel interference cancellation—demonstrates the delicate interplay between *performance*, *robustness*, and *efficiency*, a balance that guides the core contributions of this work.

Structure of the Dissertation. To delve deeper into these ideas, the dissertation is organized as follows:

- **Part I:** Investigates ML-based parameter estimation techniques, focusing on AoA estimation and how to adapt models to unseen or changing wireless environments.
- **Part II:** Explores ML-driven interference cancellation, leveraging advanced neural architectures such as U-Nets, Fast Fourier Convolutions, and LSTMs, followed by model compression strategies to enable low-latency, resource-

friendly deployment.

- **Part III:** Introduces learned image compression frameworks that aim to maintain reliability, high throughput, and low latency in the presence of unreliable channels, demonstrating progressive transmission and decoding strategies.

By addressing these problems in turn, this dissertation offers a cohesive narrative on the design, adaptation, and optimization of ML solutions for next-generation wireless and signal processing tasks.

1.4.1 Motivation

The growing complexity and demands of modern communication systems highlight a fundamental tension: on one hand, classical theories (e.g., linear signal processing, Gaussian noise assumptions) provide clear analytic frameworks and performance guarantees, but often fail to capture real-world nonlinearities and time-varying phenomena. On the other hand, machine learning (ML) offers a powerful data-driven paradigm capable of handling complex, high-dimensional data. Yet it comes with its own set of issues—namely, heavy computational requirements, sensitivity to distribution shifts, and difficulties in interpretability.

In practical scenarios such as next-generation wireless networks, IoT ecosystems, and edge computing devices, several challenges frequently arise. Rapid environmental changes cause communication links to degrade or fluctuate dramatically due to mobility or interference, rendering static models ineffective. Many end devices operate under tight power and processing limitations, which constrain computational resources and make large ML models impractical unless they are carefully optimized. Furthermore, applications like smart healthcare, autonomous vehicles, or industrial automation require high reliability and cannot tolerate frequent system failures, thus necessitating robust learning strategies. Additionally, real-world wireless data may be noisy, sparse in certain operating regimes, or highly variable, which poses significant challenges to both the model training process and reliable inference.

The work in this dissertation is motivated by the ambition to address these shortcomings head-on. By carefully integrating domain knowledge (e.g., known physical layer properties) into ML architectures, and by developing strategies for model adaptation, compression, and robustness, we can derive solutions that move beyond purely theoretical approaches. These solutions not only target state-of-the-art performance but also emphasize *scalability* and *practical feasibility*, thus providing a foundation for future communication systems where dynamic, data-driven signal processing will be indispensable.

1.4.2 Connecting the Topics

While this dissertation addresses three primary problem domains—parameter estimation, co-channel interference cancellation, and learned image compression—they share common threads that bind them into a coherent body of research. First and foremost, each problem underscores the tension between *accuracy* and *efficiency*: high-fidelity predictions or signal reconstructions typically require complex models, but real-world deployments demand swift inference on resource-limited hardware.

Moreover, all three domains highlight the challenges posed by *out-of-distribution* or dynamic environments, where a trained model may face scenarios that differ significantly from its training data. In parameter estimation, this issue manifests when AoA estimators, trained on certain channel conditions, must operate in previously unseen indoor or outdoor multipath scenarios. In interference cancellation, the presence of new interference sources or nonstationary channel conditions can invalidate previously learned assumptions. For image compression, varying channel qualities and diverse image contents demand adaptive methods that can gracefully handle changing bandwidth or error profiles.

Finally, each topic compels an exploration of *model compression* and *computational optimizations*, crucial for real-time performance. For parameter estimation and interference cancellation, rapid inference is essential; for image compression, efficient encoding and decoding pipelines reduce latency and save bandwidth in edge-centric use cases. These overlapping themes of *adaptability*, *robustness*, and *resource efficiency* form the connective tissue linking seemingly distinct use cases under the umbrella of *ML-driven wireless and signal processing*.

By exploring how these shared challenges and requirements play out in different application contexts, this dissertation illustrates the versatility and potential of data-driven methods. It demonstrates that solutions initially developed for one domain (e.g., fast and compact convolutional architectures) can be adapted or extended to another (e.g., progressive learned compression). In doing so, it offers a unified perspective on designing *robust*, *efficient*, and *transferable* ML solutions for next-generation communication and signal processing systems.

1.5 Dissertation Outline

This dissertation is structured into three major parts, each focusing on a distinct yet interconnected aspect of machine learning (ML) in wireless and signal processing applications. Following these thematic parts, a concluding chapter synthesizes the findings and outlines avenues for future research.

1.5.1 Part I: Machine Learning for Parameter Estimation

Part I examines the use of ML—specifically deep convolutional neural networks (CNNs)—for parameter estimation tasks in wireless communications. The primary application explored is AoA estimation, where data-driven approaches are compared with classical signal processing methods such as MUSIC and phase-difference estimators. In addressing the issue of domain mismatch, transfer learning and data augmentation strategies are proposed to adapt models trained in one environment to previously unseen settings. The chapters in this part discuss the fundamentals of classical estimators and their limitations in dynamic or mismatched environments, detail deep learning architectures for supervised parameter estimation with an emphasis on training pipelines and optimization, and present experimental evaluations under various propagation scenarios and polarization mismatches that showcase the robustness and accuracy of ML-based solutions. Additionally, they explore transfer learning and data augmentation approaches designed to mitigate performance degradation in out-of-distribution environments.

1.5.2 Part II: Interference Cancellation and Model Compression

Part II focuses on co-channel interference cancellation using ML-driven methods, exploring advanced neural architectures and model compression techniques. In this context, U-Net architectures for blind source separation are investigated through the use of time-frequency domain representations, fast Fourier convolutions, and hybrid layers such as LSTM bottlenecks to capture long-range dependencies in interference scenarios. Recognizing the need for real-time, on-device inference, strategies including pruning, quantization, and depthwise separable convolutions are implemented and evaluated, significantly reducing computational overhead while maintaining high performance in interference cancellation. Additionally, the proposed solutions are benchmarked against state-of-the-art interference cancellation models, demonstrating improvements in mean square error (MSE) as well as reductions in both parameter count and inference latency.

1.5.3 Part III: Learned Image Compression for Wireless Transmission

Part III transitions into the realm of image compression, where modern, learned compression schemes tailored to wireless channels are explored. In this context, deep generative models are introduced, including hyperprior-based methods and Vector Quantized Generative Adversarial Networks (VQGAN), which facilitate high-efficiency encoding and robust decoding. The work further presents progressive transmission frameworks that enable the decoding of partial transmissions, thereby reducing latency under adverse conditions and allowing flexible trade-

offs between image quality and transmission overhead. Additionally, adaptive compression strategies are examined by comparing learned image compression to standard and adaptive codecs (e.g., WebP), with a focus on trade-offs in reliability, throughput, and latency under Rayleigh fading channels. Extensive evaluations under varying SNRs demonstrate that progressive and adaptive methods can outperform conventional approaches in challenging wireless environments, making them particularly suitable for edge devices and IoT applications.

1.6 Challenges and Contributions

This dissertation aims to develop and evaluate machine learning (ML) frameworks that address pressing challenges in signal processing and wireless communications, specifically focusing on (i) robust parameter estimation, (ii) co-channel interference cancellation, and (iii) learned image compression. In doing so, it bridges theoretical insights from classical signal processing with practical considerations of computational efficiency and real-world deployment constraints.

1.6.1 Challenges

Machine learning (ML) techniques promise significant advancements in wireless communications and signal processing, yet practical deployments must address several persistent hurdles. This dissertation examines how different chapters tackle these challenges through a variety of architectures and methodological innovations. Below, we outline five key challenges and specify the chapters where they are most directly addressed.

(C1) Data Availability and Quality. Many ML algorithms rely on large, high-quality datasets to learn robust models. However, acquiring representative data in diverse or harsh wireless environments can be infeasible, and labeled data are particularly scarce for tasks such as AoA [14] estimation or blind interference cancellation. This dissertation tackles data limitations in:

- **Chapter 3:** Proposes data augmentation and transfer learning to adapt AoA models when only a few observations from a new environment are available.
- **Chapter 7:** Investigates learned image compression under noisy channel conditions, where partial or progressive decoding can mitigate the need for comprehensive labeled data at all transmission stages.

(C2) Computational Constraints. Edge and embedded devices frequently operate under strict power and memory budgets, necessitating highly efficient ML models. Deep architectures can be prohibitively large or slow if left unoptimized. Several chapters focus on computational efficiency:

- **Chapter 2:** Demonstrates a CNN for AoA estimation with only a modest ($\approx 10\%$) increase in complexity compared to MUSIC, highlighting the balance between accuracy and feasible computation.
- **Chapter 5:** Uses fast Fourier convolutions and an LSTM bottleneck to reduce kernel sizes and model depth, decreasing multiply-accumulate operations without performance loss.
- **Chapter 6:** Explores depthwise separable convolutions, quantization, and pruning to achieve high throughput with minimal resource usage, specifically targeting edge hardware constraints.
- **Chapter 7:** Considers the latency and throughput trade-offs inherent in progressive image compression, where efficient partial decoding further reduces computational demands on constrained devices.

(C3) Robustness and Generalization. Wireless channels and interference patterns can shift rapidly, making it crucial that ML models generalize across different operating conditions. Addressing domain shifts and unseen environments is central to:

- **Chapter 2:** Validates CNN-based AoA estimators under hardware imperfections and varying propagation conditions, showcasing resilience compared to classical methods.
- **Chapter 3:** Focuses explicitly on adapting AoA estimators to new environments through data augmentation and transfer learning, mitigating performance degradation.
- **Chapter 4:** Investigates robust time-frequency separation of interference signals, enabling consistent performance across diverse interference types.
- **Chapters 5 & 6:** Emphasize blind interference cancellation approaches that do not rely on prior knowledge of interference characteristics, thus improving robustness.
- **Chapter 7:** Introduces progressive and adaptive image compression mechanisms to accommodate fluctuating channel conditions and error rates.

(C4) Integration with Existing Systems. Real-world deployments often require coexistence with established communication standards, signal representations and hardware, demanding backward compatibility and easy integration. While not addressed as a primary design constraint, certain chapters discuss real-world relevance:

- **Chapter 2:** Considers hardware imperfections, facilitating integration where device performance varies.
- **Chapters 4 & 5:** Employ widely used signal representations (time-frequency, Fourier transforms) that can be mapped onto existing communication workflows with minimal disruption.

- **Chapter 7:** Compares learned compression to adaptive WebP, illustrating potential pathways to blend novel techniques with established image formats and channel coding practices.

Table 1.1 summarizes these challenges and indicates which chapters primarily address each one.

Table 1.1: Overview of Challenges and Corresponding Chapters

Chapter	**(C1)**	**(C2)**	**(C3)**	**(C4)**
Ch. 2: AoA Estimation (DCNN)	✓	✓	✓	(✓)
Ch. 3: AoA in Unseen Envs	✓		✓	
Ch. 4: Time-Frequency Interference			✓	✓
Ch. 5: Blind CCI w/ FFT Convs		✓	✓	✓
Ch. 6: High-Throughput CCI		✓	✓	
Ch. 7: Learned Image Compression	✓	✓	✓	(✓)

Notes: **(C1)** Data Availability and Quality, **(C2)** Computational Constraints, **(C3)** Robustness and Generalization, **(C4)** Integration with existing Systems. "(✓)" indicates partial or indirect coverage of the challenge.

By situating each chapter in the context of these challenges, this dissertation demonstrates how cutting-edge ML architectures and strategies can be systematically tailored to overcome the limitations of classical methods and push the boundaries of performance, efficiency, and adaptability in next-generation wireless communication systems.

1.6.2 Contributions

By addressing the above objectives, this dissertation makes several contributions to the state of the art:

Contribution 1: A deep learning-based approach to parameter estimation that outperforms classical estimators (e.g., MUSIC, phase-difference) in terms of accuracy, particularly in complex or mismatched environments. This includes transfer learning and data augmentation strategies to adapt pretrained models to unseen domains with minimal additional data.

Contribution 2: Novel U-Net architectures for blind co-channel interference cancellation that incorporate time-frequency domain representations, fast Fourier convolutions, and LSTMs to capture both local and global signal dependencies. Comparative studies demonstrate significant improvements in mean square error (MSE) relative to baseline models.

Contribution 3: Demonstrations of efficient inference via model compression—quantization, pruning, and depthwise separable convolutions—that reduce compu-

tational complexity and memory footprint, enabling near-real-time performance on edge devices. Detailed evaluations highlight the trade-off between model size and interference cancellation or estimation accuracy.
Contribution 4: Progressive learned image compression schemes that adapt to varying channel conditions, maintaining high-quality reconstruction while reducing latency. These methods incorporate advanced architectures (e.g., hyperprior models, VQGAN) and dynamic transmission policies to preserve throughput and reliability under bandwidth or error constraints.
Collectively, these contributions advance the frontiers of ML-driven wireless signal processing by offering robust, resource-conscious methods that align with the rigorous demands of next-generation communication systems.

1.7 Publications

The research results obtained during this PhD have been published in scientific journals and presented at international conferences. Below is an overview of these publications:

1.7.1 Publications in international journals (listed in the Science Citation Index) [1]

1. **M. Naseri**, A. Shahid, G. J. Gordebeke, S. Lemey, M. Boes, S. Van de Velde, E. De Poorter. *Machine learning-based angle of arrival estimation for ultra-wide band radios*. Published in IEEE Communications Letters, Volume 26(6), 2022, pages 1273–1277. **(Q1, IF: 3.7)**

2. **M. Naseri**, A. Shahid, E. De Poorter. *Adapting UWB AoA estimation towards unseen environments using transfer learning and data augmentation*. Published in Internet of Things, Volume 27, 2024, page 101298. **(Q1, IF: 6.0)**

3. **M. Naseri**, P. Ashtari, M. Seif, E. De Poorter, H. V. Poor, A. Shahid. *Deep Learning-Based Image Compression for Wireless Communications: Impacts on Reliability, Throughput, and Latency*. Submitted to IEEE Journal on Selected Areas in Communications, 2025. **(Q1, IF: 13.8)**.

1.7.2 Publications in other international journals

1. **M. Naseri**, E. De Poorter, I. Moerman, H. V. Poor, A. Shahid. *High-Throughput Blind Co-Channel Interference Cancellation for Edge Devices*

[1] According to Ghent University, 'A1 publications' are articles listed in the Science Citation Index Expanded, the Social Science Citation Index, or the Arts and Humanities Citation Index, restricted to contributions listed as article, review, letter, note or proceedings paper.

Using Depthwise Separable Convolutions, Quantization, and Pruning. Published in IEEE Open Journal of the Communications Society, 2024. **(Q1, IF: 6.3)**.

2. H. Navidan, **M. Naseri**, I. Moerman, A. Shahid. *Radio Resource Management for Intelligent Neutral Host (INH) in Multi-Operator Environments*. Published in IEEE Open Journal of the Communications Society, 2024. **(Q1, IF: 6.3)**.

1.7.3 Publications in international conferences (listed in the Science Citation Index) [2]

1. **M. Naseri**, J. Fontaine, I. Moerman, E. De Poorter, A. Shahid. *A U-Net architecture for time-frequency interference signal separation of RF waveforms*. Published in the Proceedings of the 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), April 2024.

2. **M. Naseri**, E. De Poorter, I. Moerman, H. V. Poor, A. Shahid. *Blind Co-Channel Interference Cancellation Using Fast Fourier Convolutions*. Published in 2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring), June 2024, pages 1–2. IEEE

[2] 'P1 publications' are proceedings listed in the Conference Proceedings Citation Index – Science or Social Science and Humanities of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper, except for those already classified as A1.

References

[1] R. Eberhart and R. Dobbins. *Early neural network development history: the age of Camelot*. IEEE Engineering in Medicine and Biology Magazine, 9(3):15–18, 1990. doi:10.1109/51.59207.

[2] L. Dai, R. Jiao, F. Adachi, H. V. Poor, and L. Hanzo. *Deep Learning for Wireless Communications: An Emerging Interdisciplinary Paradigm*. IEEE Wireless Communications, 27(4):133–139, 2020. doi:10.1109/MWC.001.1900491.

[3] S. H. Shetty, S. Shetty, C. Singh, and A. Rao. *Supervised machine learning: algorithms and applications*. Fundamentals and methods of machine and deep learning: algorithms, tools and applications, pages 1–16, 2022.

[4] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia. *Machine learning for wireless communications in the Internet of Things: A comprehensive survey*. Ad Hoc Networks, 93:101913, 2019.

[5] C. Janiesch, P. Zschech, and K. Heinrich. *Machine learning and deep learning*. Electronic Markets, 31(3):685–695, 2021.

[6] I. D. Mienye, T. G. Swart, and G. Obaido. *Recurrent neural networks: A comprehensive review of architectures, variants, and applications*. Information, 15(9):517, 2024.

[7] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed. *A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU*. arXiv preprint arXiv:2305.17473, 2023.

[8] A. Vaswani. *Attention is all you need*. Advances in Neural Information Processing Systems, 2017.

[9] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. *A survey on contrastive self-supervised learning*. Technologies, 9(1):2, 2020.

[10] Z. Hou, M. Qin, F. Sun, X. Ma, K. Yuan, Y. Xu, Y.-K. Chen, R. Jin, Y. Xie, and S.-Y. Kung. *Chex: Channel exploration for cnn model compression*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12287–12298, 2022.

[11] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang. *A survey on model compression for large language models*. Transactions of the Association for Computational Linguistics, 12:1556–1577, 2024.

[12] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang. *Deep Learning in Physical Layer Communications*. IEEE Wireless Communications, 26(2):93–99, 2019. doi:10.1109/MWC.2019.1800601.

[13] S. K. Thillaigovindhan, M. Roslee, S. M. I. Mitani, A. F. Osman, and F. Z. Ali. *A comprehensive survey on machine learning methods for handover optimization in 5g networks*. Electronics, 13(16):3223, 2024.

[14] F. Zafari, A. Gkelias, and K. K. Leung. *A survey of indoor localization systems and technologies*. IEEE Communications Surveys & Tutorials, 21(3):2568–2599, 2019.

Part I

Machine Learning for Parameter Estimation

2

Machine Learning-Based Angle of Arrival Estimation for Ultra-Wide Band Radios

This chapter presents a detailed study on the feasibility and benefits of employing deep convolutional neural networks (DCNNs) for ultra-wideband (UWB) angle-of-arrival (AoA) estimation, with particular emphasis on achieving high accuracy in the presence of hardware imperfections. The investigation uses a uniform linear array comprising four antenna elements and proposes a DCNN-based estimator, against which two traditional methods—the MUSIC algorithm and phase difference of arrival estimators—are benchmarked. These comparisons encompass diverse propagation conditions, varying numbers of channel impulse responses, and the presence of polarization mismatches, allowing the analysis to address both the absolute estimation error and computational complexity. A noteworthy outcome is the significant reduction in AoA estimation error, which reaches up to 80° relative to conventional approaches, achieved at only a modest increase of around 10% in computational complexity compared to MUSIC. This highlights the strong potential of advanced deep learning strategies to handle practical challenges arising from typical radio frequency hardware constraints, including variations in device performance. Furthermore, unlike prior investigations relying on empirical data from single- or dual-antenna setups, the approach described here is validated with a four-element array, underscoring its flexibility and the potential for extension to an even broader range of array configurations. Such extensibility is essential

for future applications in large-scale or heterogeneous antenna systems, where the capacity to adapt to more complex hardware layouts can be critical. Finally, this work pays special attention to the computational requirements of the DCNN-based estimator, quantifying the trade-offs between complexity and error reduction, which is particularly relevant for real-world deployments in resource-constrained environments.

Chapter 2 introduces our first DCNN-based strategy for ultra-wideband AoA estimation, establishing its robustness against hardware imperfections and showing clear benefits over conventional methods. Building on these findings, Chapter 3 extends the approach to unseen environments by integrating transfer learning and data augmentation, thus enabling the model to adapt to changing propagation scenarios with limited additional data. In doing so, the insights from Chapter 2 provide the groundwork for a generalized, environment-agnostic framework in Chapter 3.

⋆ ⋆ ⋆

This Chapter is adapted from:

Mostafa Naseri, Adnan Shahid, Gert-Jan Gordebeke, Sam Lemey, Michiel Boes, Samuel Van de Velde, Eli De Poorter
Machine Learning-Based Angle of Arrival Estimation for Ultra-Wide Band Radios

Published in IEEE Communications Letters, Volume 26, 2022.

This chapter targets C1–C3 fully and C4 partially.

2.1 Introduction

Ultra-wideband (UWB) has become a key technology for localization systems in GPS-denied environments [1], [2]. The UWB technology benefits from a high time-domain resolution leading to a precise time of flight (ToF) and high-resolution channel impulse response (CIR) measurements. The high resolution CIR provides useful information that could be used to tackle main localization challenges, e.g. multipath propagation.

The UWB technology enables several localization approaches, among which angle of arrival (AoA) estimation is highly demanded. AoA estimation is a crucial task in narrow beam wireless data transmission and smart antenna systems to facilitate beamforming [3], vehicle to vehicle communication [4], and indoor positioning [5]. Unlike approaches that require two-way communication between

anchor node and tag node, e.g. two-way ranging, in AoA estimation a feedback link is not required (in self localization) which results in better system scalability and complexity. In addition, current UWB positioning systems typically use timing information to determine the distance between a mobile tag and several distributed anchor nodes. By adding an additional antenna and radio module on the anchor node (e.g. creating an antenna array), the phase and arrival time can be determined at each antenna element, enabling the extraction of angle-of-arrival information. Hence, using AoA, the total required infrastructure cost can be reduced significantly.

Traditional AoA estimation methods are divided into several categories, namely spectral-based estimation, deterministic parameter estimation, and subspace-based AoA estimation [6]. These methods are vulnerable to array imperfections caused by suboptimal antenna design, fabrication imperfections, inter-antenna interference, and installation platform effects. In addition, challenging environmental effects, e.g. multipath and non-line of sight (NLoS), will degrade the performance of these traditional methods [7]. Modeling all of the aforementioned destructive effects is not necessarily an efficient approach, if not impractical.

As opposed to the rule-based algorithms, a deep convolutional neural network (DCNN) is adopted in this work. DCNNs select features from the input without manual feature extraction and finds a mapping from the features in the observed data to the desired output, i.e. true AoA. DCNNs have demonstrated excellent performance in the image processing domain [8]. The DCNN can overcome modeling complications of the aforementioned antenna array imperfections and extract features from the antenna array output to make the algorithm robust against environmental changes. It should be noted that relying on simulation results for evaluating machine learning (ML) solutions could be misleading due to the fact that in the simulations, the interfering effects are artificially added to the system using known models [9]. However, such imperfections could be unknown, difficult to measure, or too complicated to be modeled. To overcome the aforementioned challenges in rule-based approaches, in this work, we use supervised ML to estimate the AoA from a set of labeled input-output pairs.

Although almost all AoA estimation methods rely on the reception of the signal by an array of antennas (or one antenna that moves to different positions), [10] and [11] propose a single-antenna AoA estimation approach. Single antenna AoA estimation algorithms either require more complicated hardware [10] or exploit the angle-dependent property of the transmitter and receiver antenna pattern [11]. In the latter approach, the AoA estimator maps the CIR to AoA for one specific convolution of the impulse response of the transmit antenna, channel, and receive antenna. Since in this approach the estimator does not have access to differential information (between antenna elements), the mapping is environment-dependent for a fixed pair of transmit and receive antennas.

In addition to the error performance of the adopted algorithm, its complexity

plays an important role in an AoA estimation, especially for real-time applications [12]. For instance, iterative searches or singular value decomposition (SVD) increase the complexity of an algorithm and, hence, require a careful examination of the time complexity. Furthermore, lack of adequate snapshots can severely affect the AoA estimation algorithm [13] hence, evaluation of the algorithm performance in absence of adequate CIRs is of major importance to acquire insight on a system level.

In this paper, a supervised ML approach is proposed to achieve accurate AoA estimation robust against hardware imperfections. It is validated for clean, multipath, and high-noise conditions in various environments through experimental results obtained by a four-antenna uniform linear array (ULA). While prior work supported by empirical data is limited to single and double antenna setups, our estimator is extendable to any array configuration. The antenna array hardware is equipped with four synchronized Decawave DW1000 chips to extract the CIR at each antenna element. The main contributions of this paper are as follows:

- A supervised DCNN solution is proposed to improve AoA estimation error in multi-antenna-based UWB systems. The model is trained via high granularity labeled data captured in an anechoic chamber. The model takes the amplitude of the truncated CIR as input and outputs the estimated AoA.

- Comparison of different AoA estimation algorithms in terms of estimation error distribution and complexity. In particular, the proposed method is compared to the phase difference of arrival (PDoA) and multiple signal classification (MUSIC) algorithms.

- Analysis of the robustness of the proposed solution against small numbers of observed CIRs and antenna polarization mismatch and comparison with PDoA- and MUSIC-based AoA estimation.

2.2 System Description

Consider a single-antenna transmitter and a multi-antenna receiver in an environment characterized by N_{path} propagation paths arriving at the i-th receiver antenna element. For clarity, we first introduce the real-valued passband CIR, then describe how it relates to the complex-valued discrete-time representation used in practical receivers.

2.2.1 Passband Channel Model

The real-valued continuous-time CIR at the i-th antenna is given by

$$h_i(\tau) = \sum_{l=1}^{N_{\text{path}}} \alpha_{i,l}\, r_l(\tau) \otimes \delta(\tau - \tau_{i,l}), \tag{2.1}$$

where $\alpha_{i,l}$ and $\tau_{i,l}$ denote the gain and delay of the l-th path at the i-th antenna, respectively, $r_l(\tau)$ represents the received pulse shape distorted by propagation, $\delta(\cdot)$ is the Dirac delta function, and $\otimes$ denotes convolution [14].

The distorted pulse shape $r_l(\tau)$ depends on the transmitted pulse $p(\tau)$, the transmit antenna impulse response $h_{\text{TX}}(\tau)$, the propagation channel impulse response excluding antennas $h_{\text{H}}(\tau)$, and the receive antenna impulse response $h_{\text{RX}}(\tau)$. These impulse responses may be cascaded, depending on environmental and antenna assumptions [15, Sec. 4.2]. In linear time-invariant (LTI) conditions (e.g., static environments), the overall impulse response simplifies to

$$h_{\text{TX}}(\tau) \otimes h_{\text{H}}(\tau) \otimes h_{\text{RX}}(\tau). \tag{2.2}$$

2.2.2 Downconversion and Complex Baseband Representation

Practical receiver hardware does not directly sample the passband signal. Instead, the received real-valued RF passband signal is first downconverted to complex baseband by mixing with a local oscillator at carrier frequency f_c, followed by low-pass filtering. Mathematically, the resulting complex baseband CIR is related to the original passband CIR by

$$h_i^{\text{BB}}(\tau) = \left[h_i(\tau) \otimes e^{-j2\pi f_c \tau}\right]_{\text{LPF}}, \tag{2.3}$$

where the notation $[\cdot]_{\text{LPF}}$ indicates low-pass filtering, removing higher-frequency components. This step transforms the real-valued passband channel into a complex-valued baseband channel, explicitly representing amplitude and phase variations.

2.2.3 Discrete-Time Complex CIR and I/Q Sampling

After downconversion, the complex baseband signal is sampled at intervals of T_s seconds (sampling frequency $f_s = 1/T_s$), yielding discrete-time complex samples. Consequently, the practical discrete-time channel impulse response at the i-th antenna element is given by sampling the complex baseband CIR:

$$h_i[n] = h_i^{\text{BB}}(nT_s), \quad n \in \mathbb{Z}. \tag{2.4}$$

In practice, fractional delays ($\tau_{i,l}$ not aligned to the sampling grid) are typically accounted for using oversampling or interpolation techniques. The resulting

discrete-time CIR vectors $h_i[n]$ are complex-valued and directly correspond to the I/Q samples produced by hardware ADCs, which form the basis of subsequent digital signal processing.

For notational simplicity throughout the remainder of this paper, we retain the discrete-time complex CIR as $h_i[n]$, implicitly understanding that it arises from the complex baseband downconversion of the original passband channel described in (2.1) and (2.3). Dependencies on angles of departure and arrival or polarization are omitted from notation but implicitly present in realistic scenarios.

2.2.4 Dataset Generation and Description

The hardware setup, shown in fig. 2.1, consists of one UWB Pozyx developer tag with an on-board chip antenna [16], and one anchor board containing four DW1000 chips interconnected to an ultra-wideband 1×4 air-filled substrate-integrated-waveguide (AFSIW) cavity-backed slot antenna array [17]. The DW1000 chips are synchronized by a common clock and the antenna array is interfaced to the PCB.

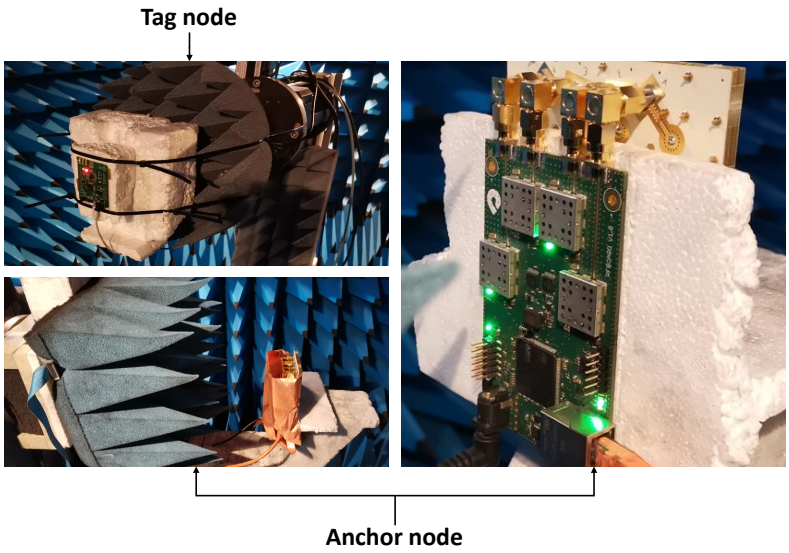


Figure 2.1: Experimental setup in an anechoic chamber: a single-antenna tag communicates with an anchor node equipped with four synchronized DW1000 chips and a 1×4 antenna array.

The experiments are conducted in clean, noisy, and multipath environments. The distance between the anchor and tag node is 2.14 m, and the transmitter has a power level of 11.5 dBm and a bandwidth of 499.2 MHz. In the clean environment experiment, no additional interfering effect is added to the anechoic chamber. In the noisy environment experiment, an extra -10 dB attenuator is added between

each antenna/DW1000 pair. Moreover, in the multipath experiment, a reflector surface is used to introduce a second path (LoS and NLoS) to the existing medium. The anechoic chamber is equipped with an NSI-700S spherical near-field scanner suitable for high granularity AoA measurements [18]. 500 CIRs are captured per AoA for [-90°, 90°] with a granularity of 1°. It is worth mentioning that since we use a hemispherical antenna with a high front-to-back ratio to spatially filter away waves impinging from the antenna backside, we choose [-90°, 90°] as the range of interest.

The multi-antenna anchor node communicates the complex-valued samples of the accumulated CIR, FP index of CIR, and synchronization frame delimiter (SFD) at the four DW1000 chips over an Ethernet interface. A CIR is constructed by accumulating the cross-correlation of the arriving preamble sequence and a known sequence with a resolution of 1 ns at each DW1000 chip. The accumulated CIRs have a length of L_{CIR} = 1016 and one can extract the phase of arrival by calculating the phase of the complex-valued CIR at the first path (FP) index. Due to the limited resolution of the CIR, i.e. 1 ns, the FP index may not report the exact arrival moment and as a result, it is recommended that the phase of arrival is calculated by averaging the phase over the FP index and its adjacent samples [10]. Note that the CIR reception rate is not constant and depends on the quality of the received signal. In our experiment setup, 53 ms is required to capture one CIR.

2.2.5 AoA Estimation of UWB Array

In this section, we formulate the AoA estimation problem taking into account the limited training and calibration data, and estimation update rate. The mean absolute error (MAE) of estimates can be defined as follows:

$$MAE(\theta) = \frac{1}{N}\sum_{i=1}^{N}|\theta - \hat{\theta}_i| \tag{2.5}$$

$$\hat{\theta}_i = f(\boldsymbol{g_1^i},\ldots,\boldsymbol{g_{N_A}^i}) \tag{2.6}$$

where N is the number of available estimates per AoA, N_A is the number of antennas in the array, $\theta \in \Theta$ = {-90, -89,. . . , 90}, $f(\cdot)$ represents the estimation algorithm, and $\boldsymbol{g_j^i}$ contains N_{CIR} complex-valued CIRs, and the corresponding SFDs and FP indices at antenna j. The estimator uses N_{CIR} CIRs, SFDs, and FP indices to generate one update (new estimate). The estimator can use N_{train} of the angles in Θ for training or calibration purposes.

2.2.6 Rule-Based Methods

In this work, the proposed DCNN-based method is compared with two rule-based AoA estimation methods, MUSIC, and PDoA. MUSIC is a spectral-based AoA

estimation method that uses the covariance matrix of the received signal $\hat{R} = \frac{1}{N_{CIR}} \sum_1^{N_{CIR}} y[n]y[n]^H = U_n \Lambda_n U_n + U_s \Lambda_s U_s$, where y is the array output, U_s and U_n are the signal and noise subspaces, Λ_s and Λ_s are diagonal matrices, and $(\cdot)^H$ is the Hermitian transpose. The MUSIC AoA estimate is equivalent to finding the direction that maximizes the MUSIC spectrum given by

$$P(\alpha) = \frac{a(\alpha)^H a(\alpha)}{a(\alpha)^H U_n U_n^H a(\alpha)} \tag{2.7}$$

where α is the direction of arrival and $a(\cdot)$ is the steering vector [6]. PDoA AoA estimation works based on the difference between the phase at adjacent antennas with a separation below half wavelength. The AoA estimate is given by

$$\hat{\theta} = \frac{1}{\beta} \arcsin \frac{\gamma}{\pi} \tag{2.8}$$

where γ is a function of PDoA and SFD of the two receivers, and β depends on the antenna array (in this work $\beta = \frac{1}{0.87}$) [19]. Note that since PDoA is only applicable to two antennas, we average three AoA estimates obtained from each subarray of two adjacent antennas in the 1×4 antenna array.

2.2.7 Calibration of MUSIC and PDoA

In our experiments, we observed that without calibration, the AoA estimates from both the MUSIC and PDoA methods were highly inaccurate. This led us to implement a calibration method to correct for phase offsets between the array elements.

Both MUSIC and PDoA rely on the phase of the received signal at the moment of arrival to each element of the array. Empirically, we observed that instead of using the phase solely at the index of the peak of the channel impulse response (CIR), averaging it with the phases from the adjacent indices improved the estimation accuracy. Consequently, we adopted this averaged phase as the input for both methods.

To perform the calibration, assume a calibration dataset $\mathbf{x}^{(m)} = [x_1^{(m)}, \ldots, x_N^{(m)}]^T$, where $\mathbf{x}^{(m)} = [x_1^{(m)}, \ldots, x_N^{(m)}]^T$ is the received signal vector at the array, and $\theta^{(m)}$ is the known true AoA for sample m. The phase offsets between the N array elements are denoted as $\phi = [\phi_1, \phi_2, \ldots, \phi_N]$, with $\phi_1 = 0$ set as the reference.

We used a sequential calibration approach, determining ϕ_k for each element $k = 2, 3, \ldots, N$ by minimizing the mean absolute error (MAE) over the subarray of elements 1 to k, with $\phi_1, \ldots, \phi_{k-1}$ fixed from prior steps.

For each element k, the corrected signals for the subarray are $\tilde{x}_i^{(m)} = x_i^{(m)} e^{-j\phi_i}$, for $i = 1, \ldots, k$. The AoA estimate $\hat{\theta}_{1:k}^{(m)}(\phi_k)$ is then computed using the respective method (MUSIC or PDoA) on the corrected subarray signals. The optimal phase

correction is found by

$$\phi_k^* = \arg\min_{\phi_k} \frac{1}{M} \sum_{m=1}^{M} \left| \hat{\theta}_{1:k}^{(m)}(\phi_k) - \theta^{(m)} \right|,$$

determined via a grid search over $\phi_k \in [-\pi, \pi)$.

After calibrating all elements, the full array signals are corrected as $\tilde{x}_i^{(m)} = x_i^{(m)} e^{-j\phi_i^*}$ for $i = 1, \ldots, N$, and the AoA is estimated using the calibrated phase offsets. For detailed implementations of MUSIC and PDoA, refer to subsec. 2.2.6 , respectively.

2.2.8 ML-Based AoA Estimation

Table 2.1 gives an overview of the proposed algorithm. The network estimates the AoA using the magnitude of windowed CIRs received by each antenna element. The magnitude of each CIR is windowed over the [FP-5, FP+30] index range. The optimal range depends on the interests in the multipath components of the CIR in the specific application. Our evaluations show that although the CIR phase conveys important information about the AoA, our proposed method achieves the same performance in terms of MAE using only the CIR's magnitude; thereby reducing the number of parameters in the network.

The proposed CNN uses two 1D convolutional layers, each followed by a rectified linear unit (RELU), a max pooling, and a dropout layer. Convolutional layers are used to detect features in their input vector, RELU is an activation function that introduces non-linearity to the network, max pooling down samples the features by choosing maximum among the adjacent values, and dropout randomly sets input units to zero during the training to avoid overfitting. A similar combination of layers is adopted in layers 6 and 7 as reported in table 2.1, and the output is flattened to form the input to the dense layers. We avoid providing mathematical formulas for each layer due to the limited length of this letter. The CNN+FC architecture was chosen for its efficiency in extracting features from CIRs and performing AoA regression, balancing performance and complexity over alternatives like RNNs or pure FC networks.

A total number of 90500 CIRs are used for training, validation, and test of the network. Instead of randomly selecting from a pool of CIR samples for test, train, and validation, we divide Θ into three subsets, i.e. Θ_{test}, Θ_{train}, and $\Theta_{validation}$. This ensures the proposed DCNN has not observed any CIR from the directions selected for testing the proposed method. Furthermore, test, train, and validation consist of 15%, 70%, and 15% of the angles in Θ, respectively. The proposed DCNN has 439,329 trainable parameters, and Adam optimizer is adopted for training of the network.

Table 2.1: Proposed Estimator Architecture

Layer	Type	Description	Layer	Type	Description
0	Input	35×4	9	norm	-
1	Conv1D	64×(size=3) + RELU	10	dropout	0.5
2	Conv1D	64×(size=3) + RELU	11	flatten	-
3	max pooling	size=4	12	Dense	256 + RELU
4	norm	-	13	Dense	256 + RELU
5	dropout	0.5	14	Dense	256 + RELU
6	Conv1D	32×(size=3) + RELU	15	Dense	256 + RELU
7	Conv1D	256×(size=3) + RELU	16	dropout	0.5
8	max pooling	size=4	17	Linear	Output

2.3 Experimental Results

In this section, we evaluate the performance of the proposed estimator in terms of MAE and the absolute value of error. All estimators are calibrated or trained for each test environment separately to evaluate the generalization of each algorithm to unseen AoAs. Fig. 2.2 depicts the average performance of PDoA, MUSIC, and our ML-based solution in different test environments for N_{CIR}=5. As depicted in fig. 2.2a, the average performance of our proposed method outperforms PDoA and MUSIC over the range [-90°, 90°]. Note that except reference angle of arrivals mentioned in table 2.2, other test AoAs are selected randomly. In addition to the non-linearity that both PDoA and MUSIC algorithms suffer from around extreme angles, MUSIC shows outliers and clipping to 90° and -90°. It is worth mentioning that although [19] proposes using a lookup table to overcome the nonlinearity around extreme angles for PDoA, figs. 2.2b and 2.2c show that a one-to-one relationship between the estimated AoA and true AoA does not exist for a wide range of AoAs. To further investigate the distribution of the absolute value of the estimation error, the 50th, 90th, and 99th percentiles of the absolute value of error are reported in table 2.2 for N_{CIR}=5.

The AoA estimation error is a function of true AoA due to the planar array configuration. Therefore, the error percentiles are reported for {-90, -60, -30, 0, 30, 60, 90}, showcasing that our proposed solution has a 99th percentile below or equal to 2.2°, 2.8°, and 2.2° for the clean, noisy, and multipath environments. These error percentile values for different AoAs show that, in addition to the general improvement, the performance of our proposed method is not degraded for extreme AoAs, in contrast to the more conventional rule-based methods.

The number of CIRs used to produce one AoA estimate, N_{CIR}, has an impact on the design of AoA estimation and tracking systems. For instance, using N_{CIR}=5 takes 267 ms on average in the clean environment experiment to generate one AoA estimate update. As a result, N_{CIR} determines the maximum angular maneuver velocity that the algorithm can capture, as well as the power consumption. Fig. 2.3

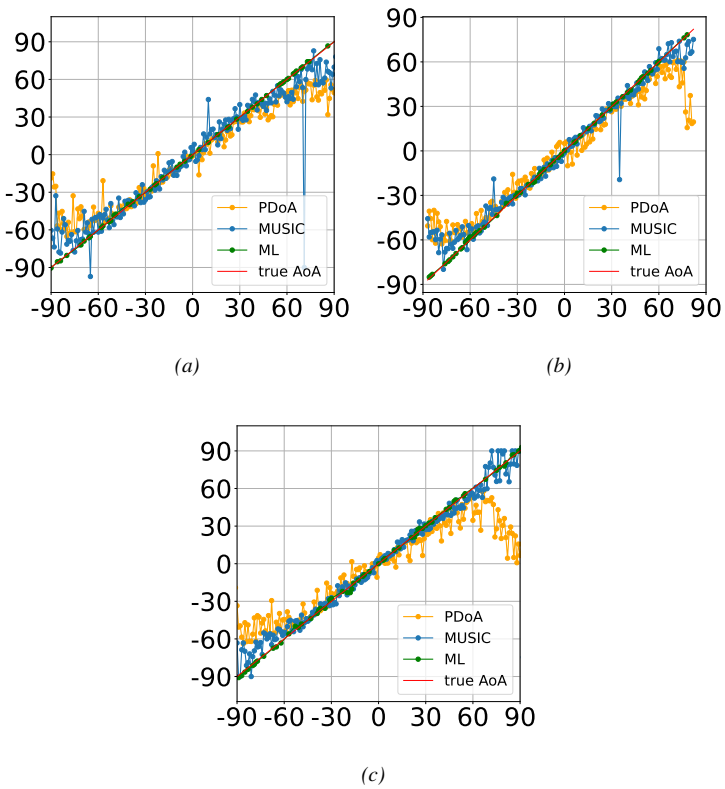


Figure 2.2: Comparison of rule-based and ML-based angle of arrival estimation in clean (a), noisy (b), and multipath (c) environments. Each scenario uses $N_{\mathrm{CIR}} = 5$ channel impulse responses per angle of arrival.

Table 2.2: Absolute Value of Error Percentiles for N_{CIR} = 5

(a) Clean Environment

AoA	MUSIC			PDoA			ML		
	50th	90th	99th	50th	90th	99th	50th	90th	99th
-90°	20.8°	32.0°	180.0°	70.2°	78.8°	80.7°	0.8°	1.3°	1.7°
-60°	5.8°	14.3°	24.6°	10.2°	18.2°	26.7°	0.4°	1.4°	2.2°
-30°	3.6°	7.7°	12.5°	1.2°	9.5°	16.9°	0.5°	1.4°	1.9°
0°	2.6°	6.9°	11.4°	1.4°	7.9°	10.36°	0.5°	0.9°	1.4°
30°	4.1°	8.6°	48.3°	1.9°	8.5°	17.6°	0.5°	1.0°	2.0°
60°	9.4°	17.6°	22.5°	9.4°	18.6°	31.9°	0.5°	1.1°	1.5°
90°	24.9°	34.8°	180.0°	58.6°	72.7°	79.6°	0.7°	1.2°	1.5°

(b) Noisy Environment

AoA	MUSIC			PDoA			ML		
	50th	90th	99th	50th	90th	99th	50th	90th	99th
-90°	31.2°	36.4°	180.0°	36.9°	51.8°	65.3°	2.2°	2.5°	2.8°
-60°	4.0°	10.1°	13.6°	6.6°	16.1°	25.2°	0.4°	1.4°	1.9°
-30°	2.1°	4.4°	5.2°	0.9°	8.6°	12.7°	0.5°	1.4°	2.4°
0°	2.0°	4.4°	6.4°	0.7°	5.6°	10.8°	0.4°	0.9°	1.5°
30°	1.7°	4.5°	7.1°	2.5°	7.8°	16.9°	0.4°	0.9°	1.0°
60°	4.2°	8.9°	20.6°	4.2°	13.0°	18.1°	0.3°	0.8°	1.0°
90°	10.1°	21.7°	180.0°	63.5°	78.6°	87.2°	0.8°	1.1°	1.4°

(c) Multipath Environment

AoA	MUSIC			PDoA			ML		
	50th	90th	99th	50th	90th	99th	50th	90th	99th
-90°	10.1°	24.5°	180.0°	65.3°	76.6°	81.8°	0.3°	0.8°	0.9°
-60°	5.4°	10.8°	13.7°	11.3°	21.2°	25.6°	0.4°	0.7°	1.2°
-30°	2.2°	6.1°	10.4°	3.0°	11.4°	16.2°	0.6°	1.7°	2.2°
0°	2.4°	4.7°	6.4°	0.9°	5.9°	10.7°	0.2°	0.4°	0.5°
30°	2.6°	6.2°	7.4°	4.5°	13.4°	22.6°	0.2°	0.3°	0.4°
60°	7.2°	11.8°	14.4°	16.5°	31.2°	36.2°	0.2°	0.4°	0.6°
90°	11.9°	22.2°	147.1°	52.5°	73.1°	84.5°	0.2°	0.4°	0.6°

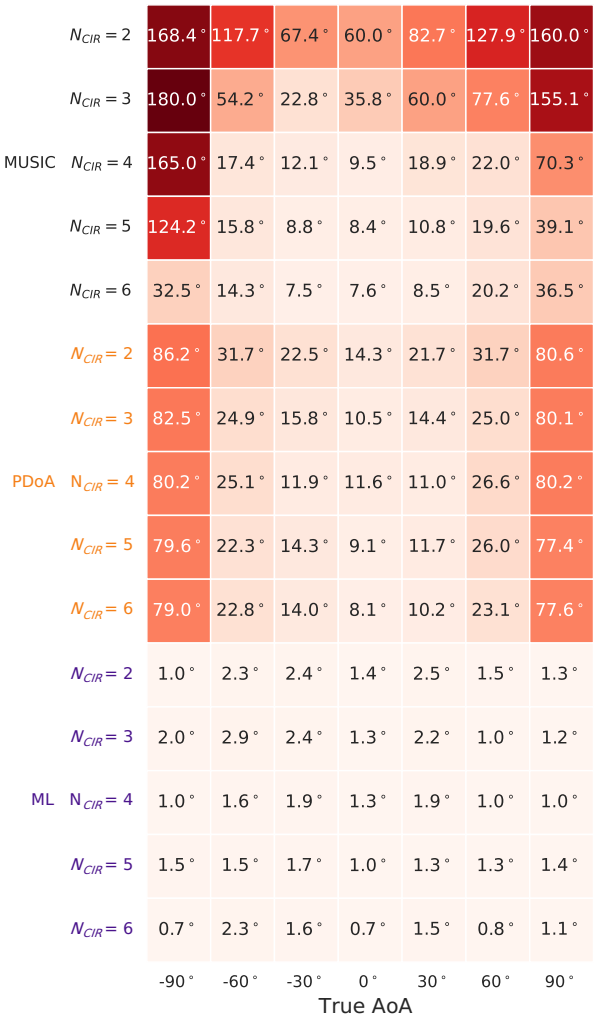


Figure 2.3: 95th percentile of the absolute angle of arrival estimation error across different N_{CIR} values in a clean environment. For each angle of arrival, an empirical CDF is constructed using 500 Channel Impulse Responses.

illustrates how N_{CIR} affects the 95th percentiles of the absolute value of error in PDoA, MUSIC, and our proposed method. For smaller values of N_{CIR}, it is more likely that MUSIC gives highly erroneous estimates, especially for extreme AoAs. PDoA on the other hand is less sensitive to N_{CIR} in terms of producing extremely erroneous estimates.

A polarization mismatch between the antenna elements in the array and the incident electromagnetic wave can distort both the amplitude and the phase of the received signal. In order to analyse the effectiveness of the proposed ML-based method against polarization mismatch, we conducted measurements for 0°, 45°, 75°, 85° polarization mismatch in the clean environment condition. Fig. 2.4 shows the average performance of the AoA estimation techniques for the aforementioned polarization mismatches. Both the MUSIC and PDoA fail to compensate for the polarization mismatch despite the fact that they were re-calibrated for each polarization independently. The proposed ML-based estimator, PDoA, and MUSIC are compared in table 2.3 in terms of average execution time for different N_{CIR}s. The proposed method has almost the same execution time as MUSIC. PDoA, however, has a shorter execution time for its simple structure. We average the execution time over 1000 runs on a device equipped with a 6 core Intel Core i7 CPU at 2.7 GHz, and 16 GB RAM using the *timeit* Python module. The total number of multiplications is $625152 = 620672 + (64 N_{ant} (L_{CIR} - 2))$ where N_{ant} and L_{CIR} are the number of antenna elements and length of windowed CIRs, respectively. Furthermore, the proposed algorithm has an average training time of is 19 minutes and 46 seconds.

Table 2.3: Average Execution Time (ms)

N_{CIR}	PDoA	MUSIC	ML
2	0.069	19.999	22.946
3	0.071	21.243	23.174
4	0.072	21.551	23.267
5	0.073	21.685	23.464
6	0.074	22.005	23.903

2.4 Conclusion

The proposed solution has a 99th percentile error of 2.8°, compared to 180° and 84.5° for MUSIC and PDoA, respectively. Furthermore, while decreasing N_{CIR} from 6 to 2 worsens the 95th percentile of the error for the traditional approaches such as MUSIC (from 36.5° to 168.4°) and PDoA (from 79.0° to 86.2°). Moreover, the proposed solution is shown to effectively solve the polarization mismatch. On

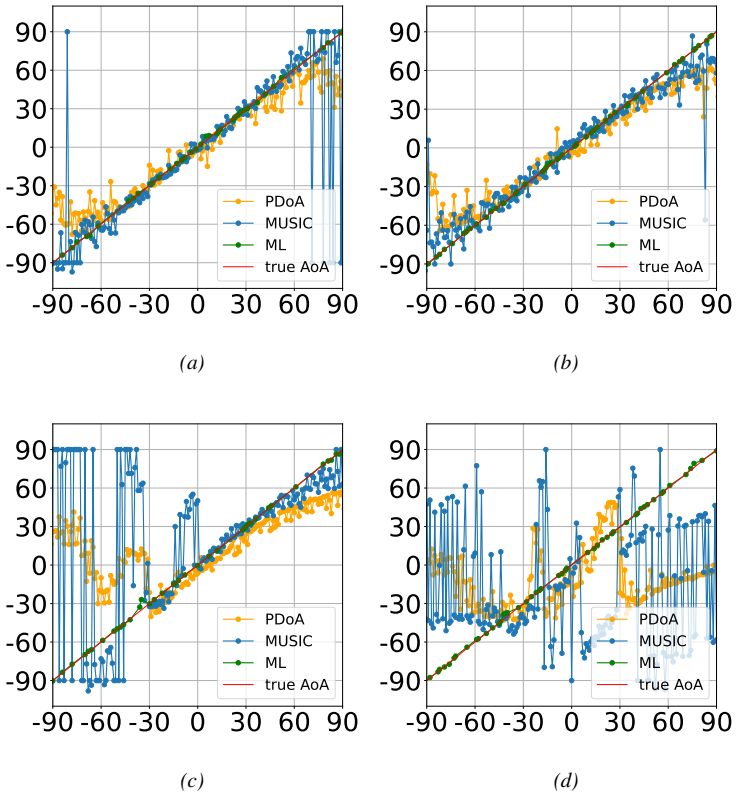


Figure 2.4: Effect of polarization mismatch on AoA estimation algorithms. The average AoA estimation performance is depicted for N_{CIR} = 5 and polarization mismatch 0° (a), 45° (b), 75° (c), and 85° (d).

the other hand, the proposed method requires more training samples as compared to calibration samples needed for rule-based methods. Finally, although the results show significant error improvements, the proposed approach has almost the same execution time as the traditional approaches. Currently, the proposed method works in the seen environment. Future work will focus on generalizability toward the unseen environments.

References

[1] A. Khan, S. Wang, and Z. Zhu. *Angle-of-Arrival Estimation Using an Adaptive Machine Learning Framework*. IEEE Communications Letters, 23(2):294–297, 2019. doi:10.1109/LCOMM.2018.2884464.

[2] F. M. Martel, J. Sidorenko, C. Bodensteiner, and M. Arens. *Augmented Reality and UWB Technology Fusion: Localization of Objects with Head Mounted Displays*. Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2018), 2018. doi:10.33012/2018.16046.

[3] M. Wang, F. Gao, S. Jin, and H. Lin. *An Overview of Enhanced Massive MIMO With Array Signal Processing Techniques*. IEEE Journal of Selected Topics in Signal Processing, 13(5):886–901, 2019. doi:10.1109/JSTSP.2019.2934931.

[4] M. Yang, B. Ai, R. He, C. Huang, Z. Ma, Z. Zhong, J. Wang, L. Pei, Y. Li, and J. Li. *Machine-Learning-Based Fast Angle-of-Arrival Recognition for Vehicular Communications*. IEEE Transactions on Vehicular Technology, 70(2):1592–1605, 2021. doi:10.1109/TVT.2021.3054757.

[5] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa. *Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances*. Sensors, 16(5), 2016. Available from: https://www.mdpi.com/1424-8220/16/5/707, doi:10.3390/s16050707.

[6] H. Krim and M. Viberg. *Two decades of array signal processing research: the parametric approach*. IEEE Signal Processing Magazine, 13(4):67–94, 1996. doi:10.1109/79.526899.

[7] S. Wielandt and L. D. Strycker. *Indoor Multipath Assisted Angle of Arrival Localization*. Sensors, 17(11), 2017. Available from: https://www.mdpi.com/1424-8220/17/11/2522, doi:10.3390/s17112522.

[8] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4):834–848, 2018. doi:10.1109/TPAMI.2017.2699184.

[9] Z. M. Liu, C. Zhang, and P. S. Yu. *Direction-of-Arrival Estimation Based on Deep Neural Networks With Robustness to Array Imperfections*. IEEE Transactions on Antennas and Propagation, 66(12):7315–7327, 2018. doi:10.1109/TAP.2018.2874430.

[10] N. Smaoui, M. Heydariaan, and O. Gnawail. *Single-Antenna AoA Estimation with UWB Radios*. In 2021 IEEE Wireless Communications and Networking Conference (WCNC), pages 1–7, 2021. doi:10.1109/WCNC49053.2021.9417526.

[11] A. Ledergerber and R. D'Andrea. *Ultra-Wideband Angle of Arrival Estimation Based on Angle-Dependent Antenna Transfer Function*. Sensors, 19(20), 2019. Available from: https://www.mdpi.com/1424-8220/19/20/4466, doi:10.3390/s19204466.

[12] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui. *Deep Learning for Super-Resolution Channel Estimation and DOA Estimation Based Massive MIMO System*. IEEE Transactions on Vehicular Technology, 67(9):8549–8560, 2018. doi:10.1109/TVT.2018.2851783.

[13] Y. Yao, H. Lei, and W. He. *A-CRNN-Based Method for Coherent DOA Estimation with Unknown Source Number*. Sensors, 20(8), 2020. Available from: https://www.mdpi.com/1424-8220/20/8/2296, doi:10.3390/s20082296.

[14] A. F. Molisch. *Ultra-Wide-Band Propagation Channels*. Proceedings of the IEEE, 97(2):353–371, 2009. doi:10.1109/JPROC.2008.2008836.

[15] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan. *Simulation of Communication Systems: Modeling, Methodology and Techniques*. Kluwer Academic Publishers, USA, 2nd edition, 2000.

[16] Pozyx. Available from: https://www.pozyx.io/.

[17] G. Singh, E. Allebes, Y. He, E. Tiurin, P. Mateman, J. F. Dijkhuis, G.-J. van Schaik, E. Bechthum, J. van Heuvel den, M. El Soussi, A. Breeschoten, H. Korpela, G.-J. Gordebeke, S. Lemey, C. Bachmann, and Y.-H. Liu. *An IR-UWB IEEE 802.15.4z Compatible Coherent Asynchronous Polar Transmitter in 28-nm CMOS*. IEEE Journal of Solid-State Circuits, pages 1–1, 2021. doi:10.1109/JSSC.2021.3116895.

[18] Electromagnetism. Available from: https://www.ugent.be/ea/idlab/en/research/research-infrastructure/electromagnetism.htm.

[19] I. Dotlic, A. Connell, H. Ma, J. Clancy, and M. McLaughlin. *Angle of arrival estimation using decawave DW1000 integrated circuits*. In 2017 14th Workshop on Positioning, Navigation and Communications (WPNC), pages 1–6, 2017. doi:10.1109/WPNC.2017.8250079.

3

Adapting UWB AoA estimation towards unseen environments

In the previous chapter, we explored the design and performance of data-driven Angle-of-Arrival (AoA) estimators for ultra-wideband (UWB) systems, comparing traditional methods (e.g., MUSIC, phase difference of arrival) with deep convolutional neural network (DCNN) approaches. While our focus there was on establishing the baseline accuracy and benefits of ML-based regression, we assumed that the training and deployment environments remained broadly consistent. In practice, however, UWB-based localization systems must contend with diverse real-world settings, where channel characteristics, multipath conditions, and hardware imperfections may differ substantially from those seen in training data. This chapter addresses the critical issue of generalization *in AoA estimation. Specifically, we present a comprehensive analysis of how DCNN, MUSIC, and PDoA perform across multiple environments, paying particular attention to the strategies required to recalibrate—or "adapt"—an existing AoA estimator to previously unseen conditions. Such adaptation might be indispensable when even moderate shifts in the physical environment cause significant degradation in estimation accuracy. Beyond merely quantifying the performance gaps, we propose a novel framework to improve DCNN-based AoA estimators in new environments. Two principal techniques are investigated:* data augmentation *and* transfer learning. *In data augmentation, synthetic or slightly altered signal observations are introduced to mimic variations from the target environment. By contrast, transfer learning fine-tunes a pre-trained DCNN with a limited number of newly collected observations. Critically, we show*

that the choice between data augmentation and transfer learning depends on the quality and quantity of available data from the new environment. Through extensive experimental evaluations, we demonstrate that our framework significantly enhances generalization, particularly in the extremes of high and low data availability. In fact, using as few as 5, 15, or 50 new observations from the unseen environment, we observe reductions in maximum generalization error of up to 83.7%, 86.8%, and 88.2%, respectively. These findings underscore the potential of hybrid ML pipelines—combining existing models with selective retraining or augmentation—to maintain robust AoA estimation in dynamically changing deployment scenarios, thereby extending the utility of data-driven localization solutions beyond their initial training contexts.

⋆ ⋆ ⋆

This Chapter is adapted from:

Mostafa Naseri, Adnan Shahid, Eli De Poorter
Adapting UWB AoA estimation towards unseen environments using transfer learning and data augmentation

Published in Internet of Things, Volume 27, 2024.

This chapter targets C1 and C3.

3.1 Introduction

3.1.1 Motivation

With the recent advent of multi-antenna ultra-wideband (UWB) radio chips, it is now possible for a single anchor node to concurrently estimate the distance between two UWB radios and the angle at which the tag is situated. This joint angle of arrival (AoA) and distance estimation using UWB technology enables localization, i.e., determining the precise position of a tag, with a single device instead of necessitating the deployment of multiple anchor nodes at various locations. This has significant implications, as it renders applications such as tracking goods and robots, localization in construction sites, and disaster management [1] [2] considerably more cost-effective due to reduced infrastructure deployment requirements.

Numerous scientific publications have already demonstrated the feasibility of achieving centimeter-level distance estimations employing UWB radios, even in challenging industrial or non-line-of-sight conditions [3] [4]. However, there is a notable lack of scientific research on enhancing the accuracy of an UWB AoA estimator under such demanding conditions.

In general, AoA estimation for UWB radios has been investigated using either (i) traditional rule-based approaches such as MUSIC and phase difference of arrival (PDoA) or (ii) machine learning (ML)-based approaches [5]. Although [5] showed that ML-based AoA estimators outperform rule-based methods, these conclusions are only valid when the neural network has been trained and evaluated in identical environments. In reality, each environment can vary, and a model trained in one environment (*train environment*) may not necessarily function accurately in another environment (*test environment*). In most realistic scenarios, node mobility and environmental dynamics necessitate a generalized AoA estimation solution. Generalization of an ML algorithm to novel conditions, e.g., to unseen environments, is a well-known problem in ML and has been studied for computer vision [6] and natural language processing [7], among other applications.

Prior work has demonstrated the effectiveness of ML when ample training data is available [5]. However, insufficient training data in AoA estimation has received less attention. Insufficient training data has been explored in the literature. For example, inadequate observations from the training environment can be addressed by artificially generating new training samples. The artificially generated data can be obtained by applying rule-based mappings to the original dataset or using ML approaches, such as generative adversarial network (GAN) [8].

Moreover, retraining pre-trained models with transfer learning (TL) can serve as a tool to tackle the issue of differing test and training environments when only a limited number of observations from the test environment are accessible. TL smartly leverages the expertise acquired in previous tasks to enhance the performance of ML solutions. TL can also improve the generalization of ML algorithms when applied to data from unseen distributions [9]. For AoA estimation, comprehensive retraining of the estimator is often impractical due to the difficulty of collecting sufficient highly granular labeled data in challenging environments. Nonetheless, akin to collecting calibration data for conventional rule-based AoA estimation algorithms, obtaining a small set of training data from the test environment appears feasible [10].

It should be noted that the aforementioned approaches, i.e., TL and rule-based artificial data generation, necessitate expert knowledge for their hyper-parameter selection. For instance, determining which layers of the pre-trained ML model should be retrained in TL or the variance of the additive noise for generating artificial data are among such hyper-parameters. In this work, we identify the hyper-parameters for transfer learning and artificial data generation through manual trial and error.

3.1.2 Related Work

In this section, we discuss the literature related to AoA estimation, as summarized in table 3.1. The majority of existing works either concentrate on single-environment UWB AoA solutions or investigate methods to enhance AoA estimation in multiple environments without considering UWB. To the best of our knowledge, our work is the first to address the generalization problem in the context of UWB AoA estimation.

Researchers have developed ML approaches for AoA estimation in various scenarios. In [5], the authors investigate the single source AoA estimation problem for UWB radios using real experimental data and propose an ML-based solution employing a convolutional neural network (CNN). The work in [11] adopts a neural network (NN) architecture comprising fully connected (FC) layers followed by one classifier branch for estimating the number of sources and four regression branches for AoA estimation. In [12], the authors propose two distinct architectures for AoA estimation, namely FC and CNN architectures. They gather data in various measurement scenarios, including non-line-of-sight (NLoS) office environment, NLoS corridor environment, and outdoor environment, as well as a combination of these scenarios as a training dataset. Consequently, the test results (with data collected from a laboratory environment) may not necessarily generalize to unseen environments. However, the study reports test outcomes for unseen signal-to-noise ratios (SNRs), i.e., higher noise levels in the test dataset compared to the noise levels in the training dataset, which is not the primary focus of this research.

In the study presented by Ref. [5], it has been demonstrated that machine learning-based solutions can significantly enhance the accuracy of AoA estimations near 90° and -90° when compared to rule-based AoA estimation techniques [13]. However, the work in [12] does not provide results for AoA estimations in proximity to 90° and -90°. This is due to the limitations in the field of view (FoV), which refers to the range of angles over which the antenna array effectively detects and estimates the AoA, of the antenna array being confined to a narrower range of AoAs, specifically, [-74°, 74°].

It is important to note that the FoV has a direct influence on the receiver hardware configuration, which sets it apart from other factors affecting generalization discussed in this research. A reduced FoV may necessitate the deployment of additional receiving antenna arrays to cover the desired range. In the present study, we investigate FoV values of $\{45°, 90°\}$, corresponding to AoA ranges of [-45°, 45°] and [-90°, 90°], bearing in mind that the selection of the FoV is contingent upon the specific application and is beyond the scope of this work.

In the context of ML input processing, the In-phase and quadrature (IQ) of captured samples can be manipulated prior to being fed into the ML model. For instance, the studies in [12] and [14] calculate the covariance matrix derived from IQ samples, which represents the correlation among different antennas in the array,

before utilizing them as input for the model. Similarly, the work in [15] computes the MUSIC spatial spectrum from IQ samples to form the ML model input.

In contrast, the studies in [16], [17], and [5] employ unprocessed complex-valued channel impulse response (CIR) as the ML model input. It is noteworthy that the use of raw IQ samples of CIR as ML model input ensures the preservation of multipath-related information. Furthermore, the high time precision of UWB technology facilitates the efficient utilization of this information.

In the literature, various ML architectures and the influence of artificial data generation on the robustness of ML solutions have been investigated. For example, in [12], the authors incorporate noise into the training dataset, sensor position, and mutual antenna coupling as a data augmentation (DA) method to develop a more robust AoA estimation algorithm. It is worth noting that each DA transformation can be executed using a specific strategy, such as the corresponding magnitude (or other parameters based on the chosen mapping) and the probability of applying the transformation. A more comprehensive study on DA techniques for time series data can be found in [18] and the references therein.

Focusing on array imperfections, such as gain and phase error, sensor position error, and mutual coupling, Ref. [19] employs an autoencoder combined with FC layers to enhance generalization to unseen array imperfection scenarios. In Ref. [20], a network is designed to learn the generation of augmented data (in this case, face images) that minimizes the target network loss. Ref. [21] introduces a search framework to automate the optimal augmentation strategy search process, utilizing a reinforcement learning controller to repeatedly sample from a strategy space, train the target network with the specific strategy, and measure its loss. Researchers in [22] concentrate on the costly search phase of an automated augmentation strategy and suggest a reduced search space for automated DA.

However, the effectiveness of various DA techniques for AoA estimation using CIR in unseen environments remains unexplored. To the best of our knowledge, this is the first study addressing generalization to unseen environments for AoA estimation using DA and TL. Table 3.1 highlights the differences between this research and previous works.

3.1.3 Contributions

This paper makes significant strides in advancing the state-of-the-art in AoA estimation for UWB systems through the following contributions:

1. We present a comprehensive analysis of the performance degradation experienced by three state-of-the-art AoA estimation techniques (PDoA, multiple signal classification (MUSIC), and ML) when applied in an unseen environment, providing insights into the robustness and adaptability of these methods.

Table 3.1: Prior work of ML-based AoA estimation in scientific literature. There are very few works focusing on the generalization of AoA estimations, and none on the generalization of UWB AoA estimators.

Reference	Technology	ML Input	Architecture	Generalization towards	TL[a]	DA[b]
[5]	UWB	CIR	CNN+FC	x	x	x
[11]	Narrow band	Array response	FC	x	x	x
[12]	Narrow band	Cov. matrix	CNN+FC	SNR	x	✓
[14]	Narrow band	Spatial spectrum	CNN+FC	x	x	x
[19]	Radio	Cov. matrix	Autoencoder+FC	Array imperfections	x	x
[23]	Microphone array	Cov. matrix+reliability metric	LSTM+FC	x	x	x
This work	UWB	CIR	CNN+FC	Unseen environment	✓	✓

[a] Transfer Learning [b] Data Augmentation

2. We introduce a novel empirical framework to quantify the minimum number of CIR samples necessary to effectively retrain AoA estimation systems for new environments, which has practical implications for the deployment of adaptive UWB systems.

The remainder of this article is structured as follows. Section II elucidates the system model and problem formulation. In Section III, the dataset is introduced, and the configuration pertaining to measurement campaigns is reported. Subsequently, Section IV expounds on the proposed algorithm employing DA and TL. Following the explanation of the proposed solution, Section V presents and deliberates on the experimental results. Lastly, Section VI offers a conclusion by encapsulating the principal discoveries of this study.

3.2 System Model and Problem Statement

In this section, the AoA estimation problem is formulated based on the system description provided earlier in sec. 2.2, explicitly considering factors that influence generalization.

We consider a linear antenna array with N_A receiver antenna elements. According to the model in sec. 2.2, the discrete-time complex-valued CIR at the n-th receiver antenna is represented as $h_n[t]$, where $n \in \{1, 2, \ldots, N_A\}$. The discrete-time complex CIR $h_n[t]$ is related to the original continuous-time CIR through baseband downconversion and sampling, as detailed in sec. 2.2.

In the context of UWB radio hardware, the discrete complex-valued CIR, denoted here as $h_n[t]$, is provided directly by the receiver's accumulator unit as a discrete sequence composed of a specific number of samples and bits. These discrete sequences form the input to the AoA estimation algorithm [24]. Formally, the AoA estimation can be represented by:

$$\hat{\theta} = f(h), \tag{3.1}$$

where $f(\cdot)$ denotes the AoA estimator, θ is the angle of arrival, and $h = [h_1, h_2, \ldots, h_{N_A}]$ is the observation matrix composed of discrete-time complex CIR samples. Here, each vector h_n is defined as $h_n = [h_n[t]]$, where $t \in \{FP - N_b, \ldots, FP + N_f\}$, with FP representing the first path index, and N_b and N_f denoting the numbers of samples before and after the first path index, respectively.

For training or calibrating AoA estimators, both ML-based and rule-based estimators require labeled datasets. To ensure a fair comparison between these two classes of estimators, identical training datasets are used [5]. Specifically, the datasets consist of (i) a training subset (H_{tr}, Θ_{tr}) for model construction, (ii) a validation subset (H_{val}, Θ_{val}) for evaluating convergence during training, and (iii) a testing subset $(H_{test}, \Theta_{test})$ for assessing final performance. Each dataset captures N_{CIR} CIRs per AoA.

The training process for an ML-based estimator $f_{ML}(\cdot)$ involves minimizing a loss function iteratively using H_{tr} and H_{val}:

$$f_{ML} \leftarrow \min_{f}(Loss(f, \{H_{tr}, H_{val}\})). \tag{3.2}$$

The loss function considered here is the mean absolute error (MAE) defined as:

$$Loss(f, H) = \frac{\sum_{h_i \in H} |\theta_i - f(h_i)|}{|H|}, \tag{3.3}$$

where θ_i is the true AoA corresponding to the observation h_i.

We propose the mean absolute generalization error (MAGE) metric to quantify the generalization performance of AoA estimators trained (or calibrated) in one environment (E_{tr}) and tested in another (E_{test}). The proposed metric is defined as:

$$MAGE(E_{tr}, E_{test}) = \frac{\sum_{h_i \in H_{test}} |\theta_i - f(H(\theta_i))|}{N_{CIR}|\Theta_{test}|}, \tag{3.4}$$

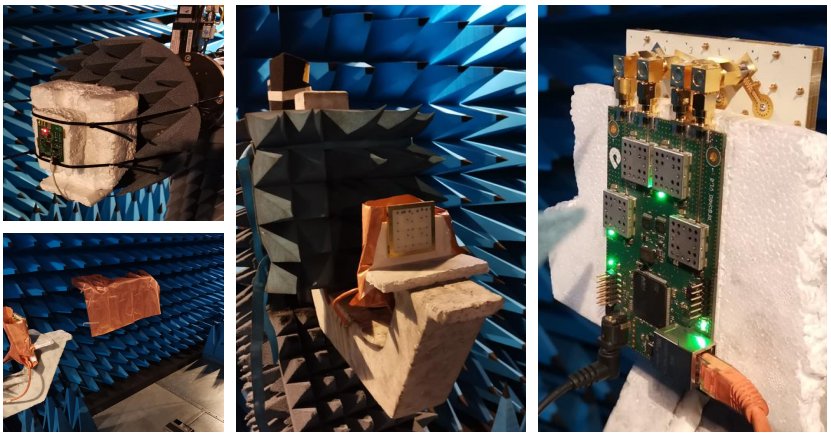
where $H(\theta_i)$ represents all CIRs measured at AoA θ_i.

Additionally, the generalization performance depends on the model complexity (C). Thus, the AoA estimator function $f(H)$ can explicitly be expressed as $f(H; P)$, where $P = [E_{tr}, N_b, N_f, N_{CIR}, FoV, C]$ denotes parameters influencing the generalization. For brevity, the dependency on P is not explicitly shown in the rest of this manuscript.

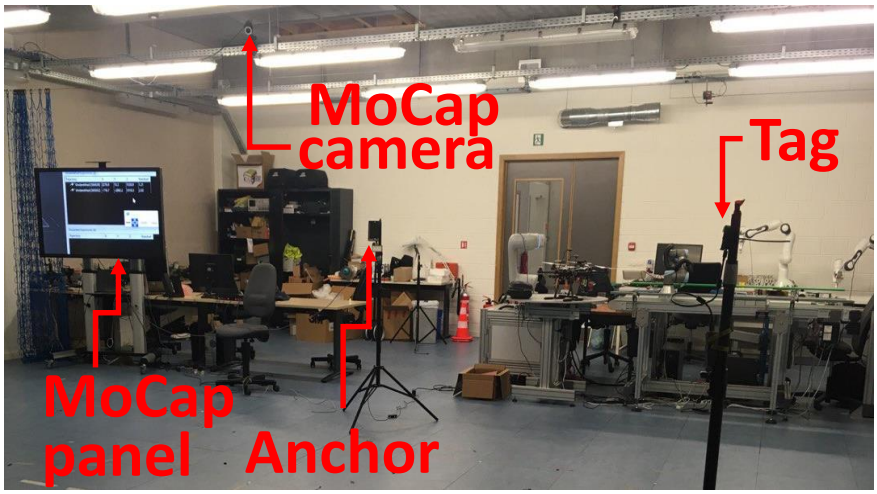
3.3 Dataset Description

In this work, three datasets are utilized, namely clean, multipath, and IIoT. The clean and multipath datasets are obtained in an anechoic chamber[1], as shown in fig.

[1] Ghent University, Electromagnetism, https://www.ugent.be/ea/idlab/en/research/research-infrastructure/electromagnetism.htm.



(a)



(b)

Figure 3.1: Measurement campaigns conducted in two environments: an anechoic chamber (a) and an Industrial Internet of Things (IIoT) setting (b). The setup includes an anchor equipped with a 1×4 antenna array and a single-antenna tag. The motion capture (MoCap) panel, camera, as well as UWB tag, and anchor are annotated in IIoT.

Table 3.2: Notations of the variables and acronyms used

Parameters	Meaning
θ	true AoA
$\hat{\theta}$	estimated AoA
τ	time index of CIR
τ_m	time delay of the m-th path
M_{path}	number of paths in the propagation channel model
FoV	range of angles antenna array detects signals from
N_A	number of antennas
h_n	complex-valued CIR at the n-th antenna
h	set of complex-valued CIRs from all antennas
α_m	complex-valued gain of the m-propagation path
r_m	distorted pulse on the m-propagation path
f	AoA estimator
P	set of parameters that impact generalization of AoA estimation
FP	first path index of CIR
N_b	number of samples before FP included in CIR
N_f	number of samples after FP included in CIR
N_{CIR}	number captured CIRs per θ
$\|\Theta\|$	number of AoAs in Θ
Θ_{tr}	set of AoAs used for training
Θ_{val}	set of AoAs used for validation
Θ_{test}	set of AoAs used for test
H_{tr}	set of CIRs used for training
H_{val}	set of CIRs used for validation
H_{test}	set of CIRs used for test
E_{tr}	training environment
E_{val}	validation environment
E_{test}	test environment
MAE	mean absolute error
$MAGE$	mean absolute generalization error
C	complexity of the model
$Flag$	a boolean variable that is $True$ only after convergence
$\mathcal{N}(\mu, \sigma)$	Gaussian distribution with mean μ and standard deviation σ
$\mathcal{U}(a, b)$	uniform distribution with minimum a and maximum b
M	number of AoAs observed from E_{test} for retraining

3.1a. For these two datasets, N_{CIR} CIRs are recorded for each AoA in the range of [-90°, 90°] with 1° granularity and 20 Hz update rate at a fixed distance of 2.14 m between the UWB pulse transmitter (tag) and the antenna array receiver (anchor). In the multipath environment, a reflecting surface is introduced to the initial clean anechoic chamber setup, causing a multipath effect on the signal propagation.

The IIoT dataset is acquired in an industrial IoT laboratory[2], depicted in fig. 3.1b. Contrary to the anechoic chamber datasets, the IIoT dataset contains channel impulse responses (CIRs) captured at variable angular steps over [-90°, 90°] with an average granularity of 1° and variable distance between the tag and the anchor. Each recorded CIR consists of 1016 complex values sampled every 10^{-9} seconds. The FP index is extracted using the leading edge detection (LDE) algorithm embedded in the DW1000. This LDE algorithm is used to detect the first path (leading edge) of the received signal and is responsible for the accuracy of proximity detection to within +/- 10 cm [25].

[2]Ghent University, Industrial IoT, https://www.ugent.be/ea/idlab/en/research/research-infrastructure/industrial-iot-lab.htm.

In structuring the dataset utilization for our models, specific considerations are made for both anechoic chamber and IIoT datasets. The anechoic chamber datasets, i.e. clean and multipath, employ a split of 80%, 15%, and 5% for training, testing, and validation respectively. It is important to note that these percentages refer to the number of AoAs. Once an AoA is selected, all N_{CIR} CIRs collected at that angle are recorded into that specific dataset. This ensures that the consistency and integrity of data from each AoA is maintained throughout the model's training and evaluation processes.

On the other hand, the IIoT dataset follows a slightly different procedure due to its character as an unseen environment in this work. Here, we use M AoAs (a value that will be determined subsequently) for the process of model retraining. The remaining AoAs, outside of these selected M, are used for the model's evaluation. This procedure offers a suitable approach for coping with the variable conditions inherent in the IIoT environment. Further details regarding the choice of M and its implications on the retraining process will be addressed in the subsequent sections of this manuscript.

Our chosen methodology, which involves assigning datasets according to AoAs rather than individual CIRs, facilitates a more comprehensive learning and evaluation phase for the models. This allocation is vital as it helps to maintain the inherent correlation and uniformity shared among CIRs from the same AoA, which could otherwise be disrupted if scattered across various datasets. By keeping these intrinsic data structures and mutual traits within the context of each AoA, our methodology ensures a more consistent and efficient training and validation process. Within the scope of our work, this approach holds significant value as it enables the model to discern and generalize the distinct CIR profiles associated with each AoA. Thus, it underlines the importance of preserving the unique AoA-specific characteristics in the data, enhancing the model's capability to deliver more accurate and robust outcomes.

To train the ML-based estimators and calibrate the rule-based estimators, true AoAs are required in addition to the complex-valued CIRs. In the anechoic chamber, a command is sent to the rotation platform to set the anchor's direction before each CIR transmission, thus establishing specific AoAs. For the IIoT measurement campaign, mm-level accuracy motion capture (MoCap) cameras [26] are employed to calculate the coordinates of the tag and the anchor, providing accurate AoA labels.

The IIoT dataset is collected in a distinct environment compared to the ideal anechoic chamber, mimicking a realistic industrial warehouse setting. This results in different AoAs (labels) and distances compared to the anechoic chamber dataset. The transmitter used in this research operates at a center frequency of 6489.6 MHz, 499.2 MHz bandwidth, and 25 dBm power level. Additionally, a pulse repetition frequency of 64 MHz, a preamble length of 64 symbols, and a data rate of 6.8 Mbps

are employed.

3.4 Proposed Method

During the training phase, ML models aim to minimize the error on the training dataset; however, the primary objective is to reduce the error on the test dataset. In this section, we introduce an algorithm designed to enhance the generalization performance of an AoA estimator under two assumptions: A) the test environment E_{test} is not previously encountered, and B) a limited number of observations from the test environment E_{test} are accessible.

3.4.1 Without Training Samples from E_{test}

We begin by considering a scenario where obtaining even a small training dataset from the target test environment, i.e., the IIoT laboratory, is not feasible. Despite this limitation, there exist methods that can improve the generalization of AoA estimation without requiring any observations from E_{test}.

One factor that can affect the generalization performance of a model is its complexity, which is determined by the number of parameters. A highly complex model may lead to overfitting and, consequently, poor generalization performance [27]. Conversely, an excessive reduction in the number of parameters of the ML model may hinder its ability to accurately model a phenomenon, such as AoA estimation. In addition to model complexity, the preprocessing of input data can also influence the generalization of the ML model. One such factor is the number of captured CIRs per AoA used during training, denoted as N_{CIR}. Although a larger N_{CIR} corresponds to more observations and reduces the MAE when E_{test}=E_{tr}, it may lead to overfitting to the current environment and result in a larger MAGE, thereby degrading generalization performance.

A potential solution to this issue is to diversify the observed CIRs, which involves expanding the captured dataset to emulate different environments. This approach is particularly useful when the available training dataset has not been collected in various environments, as it can enhance generalization by forcing the ML model to minimize its error on an extended version of the existing dataset. The extended training dataset consists of the original training data combined with artificially generated data created for diversification purposes. In this study, we employ DA to diversify the original training dataset obtained in the anechoic chamber. DA generates artificial samples by applying a set of operations to the original dataset. In this work, we propose the utilization of data augmentation techniques incorporating different forms of noise applied to the original signal $h_n[t]$. Specifically, we employ additive noise, defined by

$$h_n[t] \leftarrow h_n[t] + \text{noise}, \tag{3.5}$$

where the additive noise is a complex-valued random variable with the same dimensions as $h_n[t]$. Additionally, multiplicative noise is expressed as

$$h_n[t] \leftarrow h_n[t] \times \text{noise}, \tag{3.6}$$

where, similarly, the multiplicative noise is a complex-valued random variable with dimensions identical to those of $h_n[t]$. Finally, we apply a random shift as represented by

$$h_n[t] \leftarrow h_n[t + \text{noise}], \tag{3.7}$$

where the shift noise takes integer values, effectively altering the temporal indexing of the signal. In these equations, the symbol $\leftarrow$ denotes assignment, indicating that the value on the right-hand side is assigned to the variable on the left-hand side. The chosen DA methods reflect realistic variations in wireless signal propagation. Additive and multiplicative noise can mimic fluctuations in signal amplitude and channel noise, while the random shift emulates timing uncertainties encountered in dynamic environments.

It is important to note that, due to the characteristics of the AoA estimation problem, the time difference between CIRs received at distinct antenna elements, denoted by n, is a crucial piece of information that should not be artificially altered during the data augmentation process. Consequently, the random shift in eq. (3.7) is applied uniformly to CIRs received at all antenna elements, i.e., $n = \{1, 2, \ldots, N_A\}$.

3.4.2 Leveraging Training Samples from E_{test}

Apart from the approaches aimed at enhancing generalization performance without utilizing training samples from E_{test}, there exist methods that capitalize on incorporating observations from the novel test environment. The core concept of this subsection revolves around employing a limited set of observations from E_{test} to adapt the pre-trained ML model, thereby improving its generalization capabilities.

In this context, we employ TL to facilitate the transfer of AoA estimation expertise from a pre-trained ML model to a new estimator by retraining it with observations originating from E_{test}. This method enables the transfer of knowledge acquired in the base model to the new model by designating specific layers of the base ML model as untrainable (freezing the layers) during the retraining process. The untrainable layers execute fundamental operations pertinent to AoA estimation, which are presumed to be dataset-independent, and are thus retained in the new model. Subsequently, the remaining trainable layers are fine-tuned using the available dataset from E_{test}.

It is important to note that while the efficacy of TL is influenced by the number of accessible observations from E_{test}, employing TL proves advantageous in

comparison to training the entire model from scratch using observations from E_{test} under certain conditions, which will be elaborated upon in the results section.

Algorithm 1: Proposed generalization solution

Input:

- $H_j = \bigcup_{i \in I} H_j^i \; \forall j \in J, I = \{\text{clean, multipath, IIoT}\}, J = \{tr, val\}$
- DA Policy
- TL Policy

Output:

- $f_{ML}(\cdot)$

Step-1: Data Augmentation
for *each mapping in* DA Policy **do**
  Generate $noise$ according to DA Policy
  $H_j^i \xleftarrow{\text{mapping}} (H_j^i, noise) \; \forall i \in I, j \in J, I = \{\text{clean, multipath}\}, J = \{tr, val\}$
end
Step-2: Training the Base Model
Initialize convergence condition $Flag \leftarrow$ False
while *Flag=False* **do**
  $f_{ML}^{base} \leftarrow minimize(Loss(f_{ML}^{base}(H_{tr}, H_{val})))$
  if f_{ML}^{base} *converged* **then**
    $Flag \leftarrow$ True
  end
end
Step-3: Transfer Learning
Initialize convergence condition $Flag \leftarrow$ False
Freeze $f_{ML}^{base}(\cdot)$ layers based on TL Policy
if $|\Theta^{IIoT}| > M$ **then**
  while *Flag=False* **do**
    $f_{ML} \leftarrow minimize(Loss(f_{ML}^{base}(H_{tr}^{IIoT})))$
    if f_{ML} *converged* **then**
      $Flag \leftarrow$ True
    end
  end
else
  $f_{ML}(\cdot) \leftarrow f_{ML}^{base}(\cdot)$
end
return $f_{ML}(\cdot)$

The proposed generalization approach is delineated in a comprehensive manner in algorithm 1. This algorithm primarily takes three inputs: CIR datasets, DA strategy, and TL policy. The DA is employed for additive noise, multiplicative noise, and random shift on all input datasets. The mean and variance of the noise utilized in each DA mapping are provided in the DA policy vector. It is essential to note that DA is solely applied to the training and validation datasets, while the evaluation of the estimator is conducted exclusively on the original test dataset.

Subsequently, the augmented clean and multipath datasets are employed to train the base model f_{ML}^{base}. In this study, the number of optimizer iterations, i.e., epochs,

Table 3.3: Proposed AoA Estimator Architecture

Layer	Type	Description	Layer	Type	Description
0	input	64×8	5	dropout	0.5
1	conv1D	128×(size=3) + RELU	6	flatten	-
2	conv1D	64×(size=3) + RELU	7	dense	128 + RELU
3	max pooling	size=4	8	dropout	0.5
4	norm	-	9	output	linear

is utilized as the convergence criterion. The final stage of the proposed algorithm is contingent upon the number of AoAs observed from the test environment. Let M represent the minimum observed AoAs from the test environment for TL. It is crucial to note that if $|\Theta^{IIoT}| \leq M$, the number of observations is deemed insufficient to depend on for adjusting the weights in the f_{ML}^{base}.

3.5 Experimental Results

In this section, we assess the performance of our proposed approach and discuss the experimental outcomes with and without observations from E_{test}. A summary of the base model architecture employed in this study is provided in table 3.3. Our proposed model utilizes the architecture presented in [5] as the starting point. Through trial and error, we enhance the model's generalization and simplify it in terms of the number of parameters. The input layer of the model has a size of 64×8, with 64 samples in each CIR used in real and imaginary parts for each of the 4 antenna arrays as input to the ML model.

The model comprises two convolutional layers, each followed by a rectified linear unit (RELU) activation function, max pooling, normalization, and dropout layers before the dense layer. The convolutional layers are designed to extract features from the input signal and are followed by a RELU. The max pooling layer down-samples the features extracted in the previous layers. Subsequently, the dropout layer randomly sets its input to zero during training. Dropout is known to mitigate overfitting in the network by preventing the network from relying on a small portion of its input [28]. The flattening layer prepares the input for the dense layer, where a fully-connected layer of size 128 with a RELU activation function is employed to process the flattened values. Lastly, a linear activation function is utilized to form the estimated value.

3.5.1 Angle of Arrival Estimation in Complex Industrial Settings

Before enhancing the generalization capabilities of AoA estimation techniques, we evaluate the MAGE performance of various AoA estimators when trained

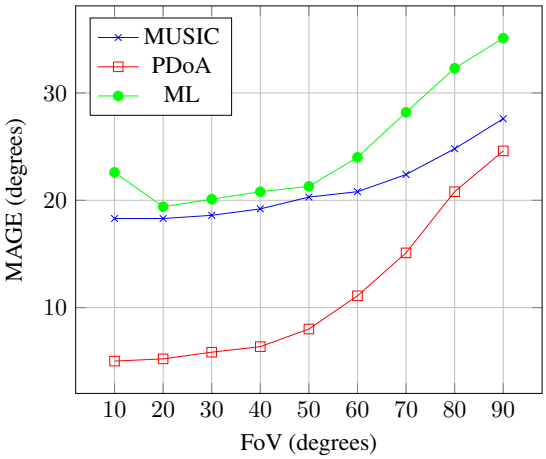


Figure 3.2: Mean Average Generalization Error (MAGE) for rule-based (PDoA, MUSIC) and machine learning-based AoA estimators across varying Field of View (FoV) angles. The increased MAGE at the edges of the FoV highlights generalization challenges in ML models and limitations imposed by the hardware.

in an anechoic chamber, i.e., on clean and multipath datasets, and tested in the IIoT environment. Fig. 3.2 illustrates the MAGE of ML, MUSIC, and PDoA AoA estimators for different FoV values. It is important to note that MUSIC and PDoA exhibit poor performance for AoAs close to 90° and -90° [5]. We consider employing rule-based AoA estimation in each anchor with smaller FoV values as an alternative generalization approach for an ML-based estimator, thus including MUSIC and PDoA in our performance comparisons. Before implementing our solution, for FoV = 90°, the MAGE is 35° for the ML AoA estimator, which is unacceptable for most applications, such as location estimation and radar [29]. Although the MAGE improves for smaller FoVs as anticipated, it does not fall below 20° even for FoVs as small as 30°. As depicted in fig. 3.2, it is clear that, without any technique for enhancing generalization, the rule-based algorithms exhibit a smaller average generalization error compared to the ML estimator, indicating a reduced degradation in average error performance. Specifically, PDoA outperforms the other two estimators, with its MAGE dropping below 10° for FoV=50°, which may be acceptable in less critical applications.

As discussed in sec. 3.4.1, P comprises parameters that influence the generalization performance of the ML-based estimator without any observation from the test environment, where $P = [E_{tr}, N_b, N_f, N_{CIR}, FoV, C]$. Regarding the complexity of the ML model C, our proposed model reduces the number of trainable parameters of the model proposed in Ref. [5] by 64%. In addition to the model's complexity, the number of captured CIRs per training AoA can result in

overfitting and, consequently, poor generalization performance. This issue can be more challenging in static environments, where multiple CIRs captured at a specific AoA may lack diversity in the feature domain that our model can potentially exploit. As a result, the model overfits the data instead of learning meaningful patterns. Fig. 3.3 (a) shows that increasing the number of captured CIRs in the anechoic chamber enhances the MAE of the model. However, the MAGE of the ML-based estimator does not follow the same trend when tested in IIoT. The minimum MAGE is achieved at $N_{CIR} = 2$, and we use this value in the rest of the analysis. It

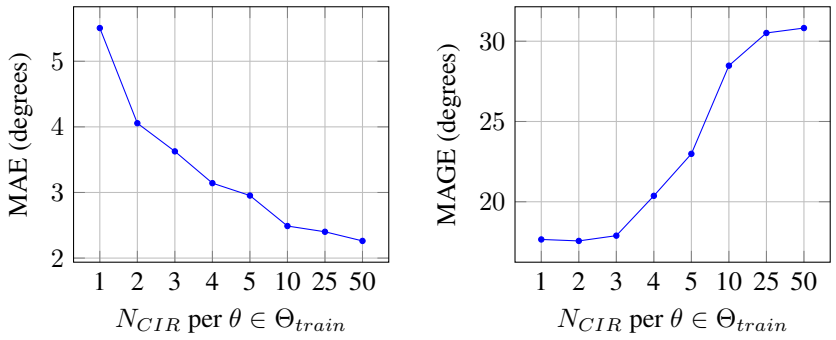


Figure 3.3: Mean Absolute Error (MAE) of the ML-based AoA estimator from Ref. [5] when the test and training environments are the same, $E_{\text{test}} = E_{\text{tr}}$ (a), and its Mean Average Generalization Error (MAGE) when they differ, $E_{\text{test}} \neq E_{\text{tr}}$ (b). In (a), both environments consist of clean and multipath conditions; in (b), the training environment includes clean and multipath, while the test environment corresponds to an IIoT scenario. Parameters: $N_b = 10$, $N_f = 54$.

should be noted that the MAGE metric may not be adequate for crucial AoA estimation tasks. Subsequently, we delve into the examination of the generalization error distribution, which offers a comprehensive statistical characterization of the generalization error. This analysis can be employed to assess various estimators for diverse applications.

3.5.2 Generalization Towards New Environments Using Data Augmentation

In this section, we present the generalization performance results when applying DA techniques to the anechoic chamber training dataset. The DA in this work employs three mappings, specifically additive noise, multiplicative noise, and random shift. The associated parameters include the mean and standard deviation of the additive and multiplicative noise, as well as the maximum of the random shift. In this article, we utilize $\mathcal{N}(0, 0.5)$, $\mathcal{N}(1, 0.02)$, and $\mathcal{U}(0, 15)$ for additive noise, multiplicative noise, and random shift, respectively. Here, $\mathcal{N}(\mu, \sigma)$ represents a

Gaussian distribution with mean μ and standard deviation σ, while for random shift $\mathcal{U}(a, b)$ denotes a discrete uniform distribution with minimum a and maximum b. It is important to note that these parameters are determined through trial and error, as finding the optimal DA policy is beyond the scope of this article.

Figure 3.4 shows the empirical cumulative distribution function (eCDF) of generalization error for machine learning (ML) and rule-based estimators using different combinations of steps of our proposed solutions. It is important to note that the reported eCDF of MUSIC and PDoA are only depicted within the error range of other ML-based models for a clear representation. In addition, MUSIC and PDoA estimators are re-calibrated using the same industrial internet of things (IIoT) observations used in transfer learning (TL) for a fair performance comparison. Calibration of MUSIC and PDoA is performed using the method explained in subsec. 2.2.7.

The blue curves in figure 3.4 show the eCDF of the ML estimator without applying DA or TL and are used as the baseline eCDF for ML-based estimators. The red curves show the eCDF when DA is applied to the baseline. DA improves the maximum of the absolute generalization error by 17.2% (2.7°) and 5.2% (0.3°) compared to the baseline for field of view (FoV) = 45° and FoV = 90°, respectively. Similar to the maximum of the generalization error, the aforementioned DA gain remains approximately constant for other percentiles of generalization error.

3.5.3 Generalization Towards New Environments Using Transfer Learning

In this section, we investigate the generalization performance of employing TL) and a combination of both DA and TL. The TL policy encompasses information concerning untrainable layers. In this study, we designate layers 0-6 from table 3.3 as untrainable and retrain the remaining layers during the TL process.

The retraining phase consists of a 50-epoch optimization, commencing with a learning rate of 10^{-3}. The learning rate decays by a factor of 0.7 following every 15 consecutive epochs without improvement in the evaluation error. Moreover, we utilize 75% of the available observations from the IIoT environment for training, while the remaining 25% serve as validation data. Given that 181 AoAs are captured in the IIoT environment, $181 - M$ AoAs remain for measuring the generalization error.

It is crucial to note that the partitioning of datasets into training, validation, and testing is based on AoA. For example, when allocating 10% of the IIoT dataset for retraining in TL, we ensure that only 10% of AoAs and their corresponding CIRs are provided to TL, rather than 10% of all CIRs, which could encompass a larger portion of AoAs. It is also worth mentioning that, besides applying DA to the anechoic chamber datasets, DA can be employed on the observed dataset from the

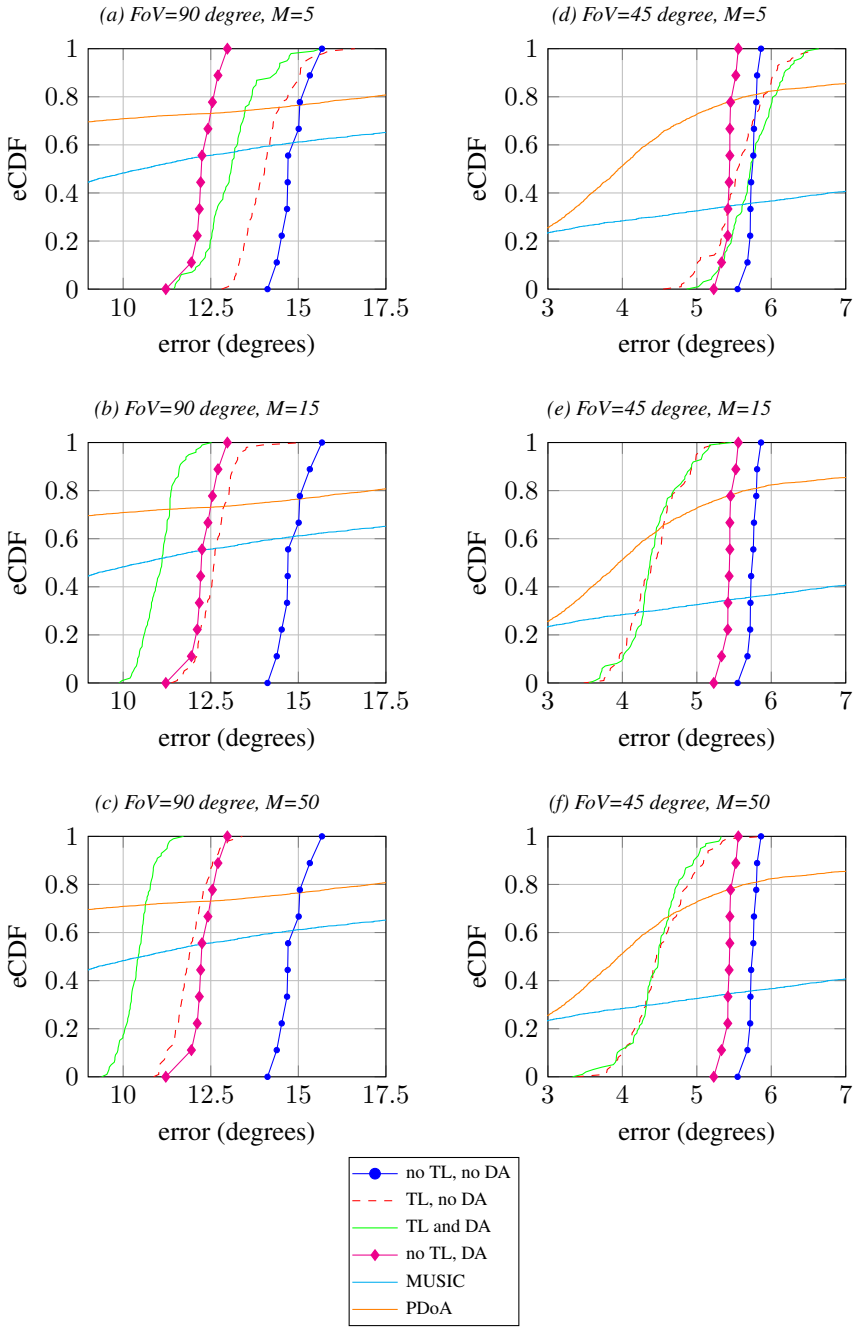


Figure 3.4: Empirical Cumulative Distribution Function (eCDF) of the absolute generalization error in transfer learning, using M Angle of Arrival samples from the IIoT environment (E_{test}). Rule-based methods (MUSIC and PDoA) are re-calibrated with IIoT data for Fields of View (FoVs) of 45° and 90°, with $N_{\text{CIR}} = 5$. The x-axis ranges from 9–17.5° (for FoV = 90°) and 3–7° (for FoV = 45°) to emphasize the main portion of the error distribution for ML-based methods, while the MUSIC and PDoA curves extend to larger error values beyond these ranges.

IIoT environment to increase its diversity. However, the test portion of the dataset is not artificially manipulated.

The eCDFs in fig. 3.4 are reported for $M = 5, 15, 50$ and FoV= $45°$,$90°$, where M denotes the number of observed AoAs from IIoT for the retraining phase during TL. In fig. 3.4, the green and orange curves represent TL with and without DA, respectively. For FoV = $90°$ and $M = 5$, the best generalization performance is achieved by applying only DA. Although TL improves the baseline eCDF (except for the maximum generalization error), it does not surpass the red curve, i.e., only DA. However, for FoV = $90°$ and $M = 15, 50$, TL enhances the eCDF performance in comparison to the only DA solution. Consequently, retraining the network using $M = 5$ samples from IIoT is inadequate, but if we have $M = 15$ or more observations from the IIoT environment, TL can improve the eCDF.

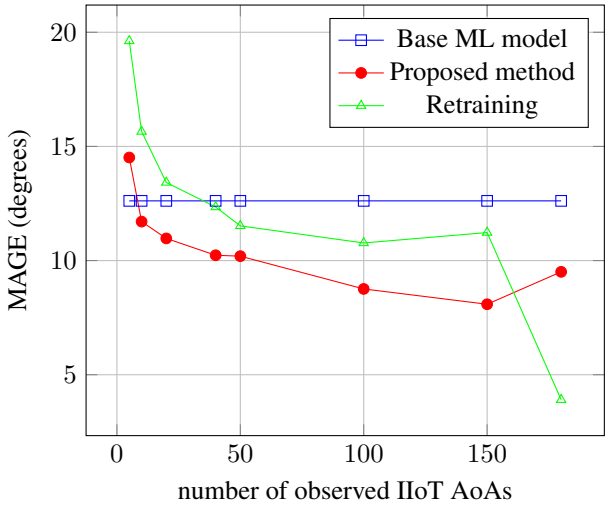


Figure 3.5: MAGE comparison for AoA estimation: improved ML method (data augmentation/transfer learning) vs. complete retraining vs. the base model. Both the proposed method and full retraining use the same number of epochs.

3.5.4 Computational Complexity Analysis

In this subsection, we discuss the computational complexity based on the number of trainable parameters, the number of multiplications, and O notation of each layer of the model.

- **Layer 0:** The initial layer is an input layer with no trainable parameters. The number of trainable parameters and multiplications are both 0.
- **Layer 1:** This is a 1D convolutional layer with 128 filters, a kernel size

of 3, and an 8-channel input. The number of trainable parameters is 3200, calculated from the weight parameters ($3 \times 8 \times 128 = 3072$) and bias parameters (128). The number of multiplications for this layer is 196608 ($3 \times 8 \times 128 \times 64$).

- **Layer 2:** Another 1D convolutional layer with 64 filters, a kernel size of 3, and 128 input channels. The number of parameters is 24640, calculated from the weight parameters ($3 \times 128 \times 64 = 24576$) and bias parameters (64). The number of multiplications for this layer is 1572864 ($3 \times 128 \times 64 \times 64$).
- **Layer 3:** This max-pooling layer does not possess trainable parameters or involve multiplications, hence number of trainable parameters and multiplications are both 0.
- **Layer 4:** This batch normalization layer has 4 parameters for each of the 64 input channels, hence has 256 (4×64) trainable parameters. Multiplication operations in this layer are not considered trainable parameters.
- **Layer 5, Layer 8:** These dropout layers do not involve trainable parameters or multiplications, hence the number of multiplications is 0 for each layer.
- **Layer 6:** This flatten layer does not have trainable parameters or involve multiplications, both are 0.
- **Layer 7:** This dense layer has 128 output units and 1024 input units. The is 131200, calculated from the weight parameters ($1024 \times 128 = 131072$) and bias parameters (128). The number of multiplications for this layer is 131072 (1024×128).
- **Layer 9:** The final layer is the output layer with 1 output unit and 128 input units. The number of trainable parameters is 129, calculated from the weight parameters ($128 \times 1 = 128$) and bias parameter (1). The number of multiplications for this layer is 128 (128×1).

The computational complexity for the layers in O notation can be represented as follows:

- *Layer 0:* $O(L_{CIR} \times 2N_A)$, where L_{CIR} and N_A are the number of samples in each CIR and the number of antennas
- *Layer 1:* $O(L_{CIR} \times 2N_A \times 64 \times 128)$
- *Layer 2:* $O(64 \times 128 \times 64)$
- *Layer 3:* $O(64 \times 64)$
- *Layer 4:* $O(16 \times 64)$

- *Layer 5:* $O(16 \times 64)$
- *Layer 6:* $O(16 \times 64)$
- *Layer 7:* $O(1024 \times 128)$
- *Layer 8:* $O(128)$
- *Layer 9:* $O(128 \times 1)$

The overall O notation for the entire model can be represented as $O(L \times 2N_A \times 64 \times 128)$.

To conclude our complexity analysis, we observe that the primary contributors to the computational complexity of the proposed model are the convolutional layers, namely *Layer 1* and *Layer 2*. In addition, the *Layer 7* layer plays a significant role in the total complexity. In contrasting the computational complexities of the MUSIC and PDoA algorithms with our proposed ML model, we observe distinct characteristics inherent to each approach. The complexity of the MUSIC algorithm is influenced by the number of antennas, N_A, with the primary computational task being the eigenvalue decomposition of the covariance matrix. This results in a complexity of $O(N_\mathrm{A}^3)$ [30], reflecting the cubic increase in computational demand as the size of the antenna array grows.

The PDoA technique, when applied to an array with N_A elements, exhibits a linear complexity, $O(N_\mathrm{A} - 1)$. This is due to the application of PDoA to each adjacent pair of antennas and averaging the results, a process that is repeated for each pair in the array. Thus, the computational complexity of PDoA increases linearly with the number of antenna elements.

Our proposed ML model, as previously detailed, has a computational complexity expressed as $O(L \times 2N_A \times 64 \times 128)$. In the context of IoT applications, where computational efficiency is crucial, understanding these complexities is key to choosing an appropriate AoA estimation technique.

3.5.5 Discussion

The Field of View (FoV) of an estimator, being a function of hardware configuration, greatly influences the estimator's overall generalization. Employing smaller FoVs necessitates a greater quantity of antenna arrays to encompass a specific angular range, thereby inflating hardware expenditure. Experimental outcomes demonstrate that PDoA surpasses MUSIC in the empirical cumulative distribution function (eCDF) of the generalization error across all investigated scenarios. In specific instances, PDoA can yield the 80th percentile of the generalization error to be 5.6° and 17.0° for FoV = 45° and FoV = 90°, correspondingly. Taking into account the relative complexity of the ML estimators to PDoA, PDoA is advised for applications where the aforementioned 80th percentile of generalization error is tolerable.

However, achieving a high-precision AoA estimation requires consideration of higher error percentiles, for instance, the 90th and 95th. By utilizing just $M = 5$ observations from the unobserved environment and applying our proposed solution, it is feasible to obtain the 90th percentile of error of 14.2° and 7.6° for FoV = 90° and FoV = 45° respectively. Fig. 3.4 portrays the advantage garnered by the proposed solution as opposed to the rule-based estimator, basic model, and partial DA and TL utilization.

The current study does not explore the impact of varying distances or hardware imperfections on the ML-based AoA estimator's performance, though designing data augmentations to model such effects could enhance robustness in future work.

In our study, the MAGE is compared for a fully retrained ML model and our proposed approach under the same number of training epochs. Fig. 3.5 demonstrates that while both methods initially underperform compared to the base ML model with minimal AoA data, the proposed method outperforms complete retraining for observation counts between 5 and 150. Notably, as observations exceed 150, complete retraining becomes advantageous, achieving lower MAGE. This finding highlights the efficacy of TL in low-data environments, aligning with our paper's focus, while acknowledging the potential of full retraining in data-abundant contexts.

3.6 Conclusion

Whereas in the past UWB localization relied mostly on distance estimations to a mobile tag, the performance can be improved by using AoA estimation in conjunction with distance estimation. Currently, three techniques are currently very popular for UWB AoA estimation: classical rule-governed AoA estimator techniques, namely PDoA and MUSIC, as well as ML based approaches. However, all of these techniques require an initial calibration.

Until now, no scientific studies focused on how existing UWB AoA estimation techniques perform when they are deployed in an environment for which they were not calibrated. Based on our experiments, we found that when the system was deployed in an unseen industrial test scenario, all three methods experienced performance deterioration. Interestingly, the decay in the effectiveness of PDoA was not as pronounced as in MUSIC and the ML approach, particularly when the field-of-view was confined to a maximum of 45 degrees. Under such constraints, PDoA displayed an average generalization error of around 6 degrees, compared to the 20-degree error typical of both MUSIC and ML strategies when no retraining of the AoA estimator was performed.

Subsequently, a novel ML architecture combined with a systematic generalization method was presented, the goal of which is to enhance the generalization capacity of ML-based AoA estimators via data augmentation or transfer learning.

From our experimental findings, we concluded that in scenarios where the quantity of obtainable samples from the new test environment falls below a certain threshold (set at 5 in this case), the best generalization performance is achieved by implementing a basic ML model trained on a broad-spectrum dataset using data augmentation exclusively. When the number of observations from the new environment is larger, it was observed that ML delivers the most optimal results when it combines both data augmentation and transfer learning. Lastly, when the availability of new samples surpasses 150, the best outcome is attained by performing a total retraining of the neural network. Using our proposed strategy, the peak AoA estimation error was seen to have improved by 83.7%, 86.8%, and 88.2% for 5, 15, and 50 observed AoAs from the novel industrial environment, respectively. Cumulatively, our enhanced ML methodology allowed the maximum AoA estimation error to consistently remain under 6 degrees in industrial settings with a field of view of 45 degrees, and under 12 degrees with a field of view of 90 degrees.

Table 3.4: 90th (a) and 50th (b) percentile error for M=15 and different FoVs

(a) 90th Percentile Error

FoV	No TL/No DA	TL/DA	PDOA
90°	15.36	11.62	29.84
45°	5.81	4.93	10.75

(b) 50th Percentile Error

FoV	No TL/No DA	TL/DA	PDOA
90°	5.74	5.20	2.30
45°	5.74	4.37	3.94

The table 3.4 underscores a trade-off between error tolerance and the need for model retraining. For the 50th percentile of error, the PDOA method suffices, eliminating the need for machine learning retraining. However, at the 90th percentile, TL and DA are critical to reduce errors significantly, necessitating the fine-tuning of ML models with new environment data for enhanced accuracy.

References

[1] A. Karaagac, J. Haxhibeqiri, M. Ridolfi, W. Joseph, I. Moerman, and J. Hoebeke. *Evaluation of accurate indoor localization systems in industrial environments*. In 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pages 1–8. IEEE, 2017.

[2] F. Zafari, A. Gkelias, and K. K. Leung. *A survey of indoor localization systems and technologies*. IEEE Communications Surveys & Tutorials, 21(3):2568–2599, 2019.

[3] F. Che, Q. Z. Ahmed, J. Fontaine, B. Van Herbruggen, A. Shahid, E. De Poorter, and P. I. Lazaridis. *Feature-Based Generalized Gaussian Distribution Method for NLoS Detection in Ultra-Wideband (UWB) Indoor Positioning System*. IEEE Sensors Journal, 22(19):18726–18739, 2022. doi:10.1109/JSEN.2022.3198680.

[4] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter. *Edge Inference for UWB Ranging Error Correction Using Autoencoders*. IEEE Access, 8:139143–139155, 2020. doi:10.1109/ACCESS.2020.3012822.

[5] M. Naseri, A. Shahid, G.-J. Gordebeke, S. Lemey, M. Boes, S. Van de Velde, and E. De Poorter. *Machine Learning-Based Angle of Arrival Estimation for Ultra-Wide Band Radios*. IEEE Communications Letters, pages 1–1, 2022. doi:10.1109/LCOMM.2022.3167020.

[6] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu. *Generalizing to Unseen Domains: A Survey on Domain Generalization*. IEEE Transactions on Knowledge and Data Engineering, pages 1–1, 2022. doi:10.1109/TKDE.2022.3178128.

[7] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf. *Transfer Learning in Natural Language Processing*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, pages 15–18, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. Available from: https://aclanthology.org/N19-5004, doi:10.18653/v1/N19-5004.

[8] P. Ganesan, S. Rajaraman, R. Long, B. Ghoraani, and S. Antani. *Assessment of data augmentation strategies toward performance improvement of abnormality classification in chest radiographs*. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pages 841–844. IEEE, 2019.

[9] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. *A comprehensive survey on transfer learning*. Proceedings of the IEEE, 109(1):43–76, 2020.

[10] M. Hussain, J. J. Bird, and D. R. Faria. *A study on CNN transfer learning for image classification*. In UK Workshop on computational Intelligence, pages 191–202. Springer, 2018.

[11] O. Bialer, N. Garnett, and T. Tirer. *Performance advantages of deep neural networks for angle of arrival estimation*. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3907–3911. IEEE, 2019.

[12] Z. Dai, Y. He, V. Tran, N. Trigoni, and A. Markham. *DeepAoANet: Learning Angle of Arrival From Software Defined Radios With Deep Neural Networks*. IEEE Access, 10:3164–3176, 2022. doi:10.1109/ACCESS.2021.3140146.

[13] I. Dotlic, A. Connell, H. Ma, J. Clancy, and M. McLaughlin. *Angle of arrival estimation using decawave DW1000 integrated circuits*. In 2017 14th Workshop on Positioning, Navigation and Communications (WPNC), pages 1–6, 2017. doi:10.1109/WPNC.2017.8250079.

[14] L. Wu, Z.-M. Liu, and Z.-T. Huang. *Deep Convolution Network for Direction of Arrival Estimation With Sparse Prior*. IEEE Signal Processing Letters, 26(11):1688–1692, 2019. doi:10.1109/LSP.2019.2945115.

[15] A. Khan, S. Wang, and Z. Zhu. *Angle-of-Arrival Estimation Using an Adaptive Machine Learning Framework*. IEEE Communications Letters, 23(2):294–297, 2019. doi:10.1109/LCOMM.2018.2884464.

[16] A. Ledergerber and R. D'Andrea. *Ultra-Wideband Angle of Arrival Estimation Based on Angle-Dependent Antenna Transfer Function*. Sensors, 19(20):4466, 2019.

[17] A. Ledergerber, M. Hamer, and R. D'Andrea. *Angle of Arrival Estimation based on Channel Impulse Response Measurements*. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6686–6692, 2019. doi:10.1109/IROS40897.2019.8967562.

[18] B. K. Iwana and S. Uchida. *An empirical survey of data augmentation for time series classification with neural networks*. Plos one, 16(7):e0254841, 2021.

[19] Z.-M. Liu, C. Zhang, and P. S. Yu. *Direction-of-Arrival Estimation Based on Deep Neural Networks With Robustness to Array Imperfections*.

IEEE Transactions on Antennas and Propagation, 66(12):7315–7327, 2018. doi:10.1109/TAP.2018.2874430.

[20] J. Lemley, S. Bazrafkan, and P. Corcoran. *Smart augmentation learning an optimal data augmentation strategy*. Ieee Access, 5:5858–5869, 2017.

[21] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. *Autoaugment: Learning augmentation strategies from data*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 113–123, 2019.

[22] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. *Randaugment: Practical automated data augmentation with a reduced search space*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 702–703, 2020.

[23] L. Cheng, X. Sun, D. Yao, J. Li, and Y. Yan. *Estimation Reliability Function Assisted Sound Source Localization With Enhanced Steering Vector Phase Difference*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29:421–435, 2021. doi:10.1109/TASLP.2020.3043107.

[24] K. Bregar and M. Mohorčič. *Improving indoor localization using convolutional neural networks on computationally restricted devices*. IEEE Access, 6:17429–17441, 2018.

[25] *DW1000 USER MANUAL*. Accessed on 08.01.2023. Available from: https://www.decawave.com.

[26] Qualisys AB. *Qualisys motion capture systems*. Accessed: 2023-5-24. Available from: https://www.qualisys.com/.

[27] P. Barbiero, G. Squillero, and A. Tonda. *Modeling generalization in machine learning: A methodological and computational study*. arXiv preprint arXiv:2006.15680, 2020.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. *Dropout: a simple way to prevent neural networks from overfitting*. The journal of machine learning research, 15(1):1929–1958, 2014.

[29] A. Broumandan, J. Nielsen, and G. Lachapelle. *Practical results of high resolution AOA estimation by the synthetic array*. In 2008 IEEE 68th Vehicular Technology Conference, pages 1–5. IEEE, 2008.

[30] R. Schmidt. *Multiple emitter location and signal parameter estimation*. IEEE Transactions on Antennas and Propagation, 34(3):276–280, 1986. doi:10.1109/TAP.1986.1143830.

Part II

Interference Cancellation and Model Compression

4

A U-Net Architecture for Time-Frequency Interference Signal Separation of RF Waveforms

In modern wireless communication systems, interference poses a persistent challenge, particularly when multiple signals overlap in the same frequency band. Accurately separating these co-channel signals is essential for preserving the signal-of-interest, which in turn facilitates enhanced demodulation and decoding. This chapter introduces a data-driven framework designed to address this critical issue by leveraging a U-Net architecture specifically tuned for time-frequency *interference signal separation.*

This paper presents a data-driven approach to solve the challenge of separating co-channel mixture signals in the radio spectrum. The main aim is to extract the signal-of-interest with high fidelity from the mixture signal, allowing improved performance in demodulation and decoding tasks. We have developed a U-Net architecture specifically designed for the separation of interference signals within the time-frequency domain. This architecture integrates elements of OFDM signal resource grid configurations, like the cyclic prefix, ensuring a tailored and effective approach to signal processing. This approach has demonstrated a significant improvement, with an average 63% enhancement in MSE performance over the baseline model on four different interference types.

Building upon traditional signal processing principles, our methodology maps time-domain waveforms to a time-frequency representation where noise and in-

terference can be more distinctly separated. The integration of OFDM-specific features, such as the cyclic prefix, further refines the U-Net's ability to address commonly encountered real-world constraints. By learning from actual interference patterns, the proposed solution achieves robust signal reconstruction, offering an average 63% improvement in mean squared error (MSE) across four separate interference categories compared to baseline models.

This chapter delves into the architectural underpinnings of our proposed U-Net, emphasizing how time-frequency processing can be adapted to extract high-fidelity signals-of-interest under diverse interference conditions. We also outline the potential implications of this work for practical communication systems, where reliable interference mitigation plays a pivotal role in maintaining quality of service. The subsequent sections detail our network design philosophy, the training procedure, and experimental evaluations demonstrating the superiority of this data-driven approach over conventional techniques.

Chapter 4 establishes our foundational U-Net approach for separating co-channel signals in the time-frequency domain, showcasing how OFDM-specific features like the cyclic prefix can be leveraged to mitigate interference. Building on these design principles, Chapter 5 extends the architecture by introducing further adaptations aimed at improving robustness and efficiency in more complex interference scenarios. Finally, Chapter 6 addresses real-time and high-throughput requirements through advanced pruning, quantization, or parallelization strategies, ensuring that the data-driven framework remains both scalable and computationally viable in practical deployments. Taken together, these chapters illustrate how data-driven architectures can be refined to deliver increasingly effective and resource-friendly interference cancellation methods.

⋆ ⋆ ⋆

This Chapter is adapted from:

Mostafa Naseri, Jaron Fontaine, Ingrid Moerman, Eli De Poorter, Adnan Shahid

A U-Net Architecture for Time-Frequency Interference Signal Separation of RF Waveforms

Published in IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), 2024.

This chapter targets C3 and C4.

4.1 Introduction

In the evolving landscape of communication systems, a critical challenge emerges from the shared use of the same segments of the radio frequency spectrum by various communication technologies. This leads to co-channel interference, a phenomenon where multiple signals overlap in the same frequency band, resulting in significant degradation of signal quality and reliability. This issue is particularly pronounced in scenarios where spectrum resources are limited [1]. Historically, the challenge of source separation in the radio frequency domain has been a topic of considerable research interest. Traditional methods such as linear minimum mean squared error (LMMSE) often rely on signal processing techniques that require knowledge of the signal characteristics or the environment. However, these methods face limitations in adaptability and scalability, especially in dynamic or complex interference scenarios.

As the demand for spectrum resources continues to grow, the need for more efficient and intelligent source separation techniques becomes increasingly important [2]. In recent years, deep learning has emerged as a powerful tool for addressing complex signal processing challenges, including the separation of overlapping signals in the RF spectrum. The ICASSP 2024 conference hosted a competition on RF signal separation, focusing on QPSK and OFDM QPSK, hereafter referred to as OFDM [3]. This paper delves into the intricacies of a U-Net[1] based architecture, exploring how it leverages the known configurations of OFDM signal grids and time-frequency representations to effectively mitigate co-channel interference.

4.2 System Model

In our study, We consider a discrete-time signal model where the received signal $y = s + b \in \mathbb{C}^L$ is a complex-valued mixture of length L. This mixture comprises the Signal of Interest (SOI) s and an interference signal b, where s and b overlap in both time and frequency. The SOI, s, is modeled as a Quadrature Phase Shift Keying (QPSK) signal shaped with a Root Raised Cosine (RRC) waveform. This setup is inspired by the configurations of the recent data-driven signal separation competition in radio spectrum: ICASSP 2024 grand challenge [4].

4.2.1 Signal Model

The n-th sample of s is expressed as:

$$s[n] = \sum_{k=0}^{L-1} a_k g[n - kF - \tau_0], \tag{4.1}$$

[1] https://github.com/MostafaUgent/UNet_time_freq_CP

where $F \in \mathbb{N}$ represents the symbol interval in discrete-time, and $\tau_0 \in \{0, \ldots, F-1\}$ denotes the offset applied to the first symbol. For our configuration, we set $F = 16$ and $\tau_0 = 8$, as suggested by the ICASSP 2024 RF signal separation challenge [3]. The function $g[n]$ is the discrete-time impulse response of the transmitter filter, which is a RRC filter with a roll-off factor of 0.5 and a window length of 127 samples, i.e. 8 symbols.

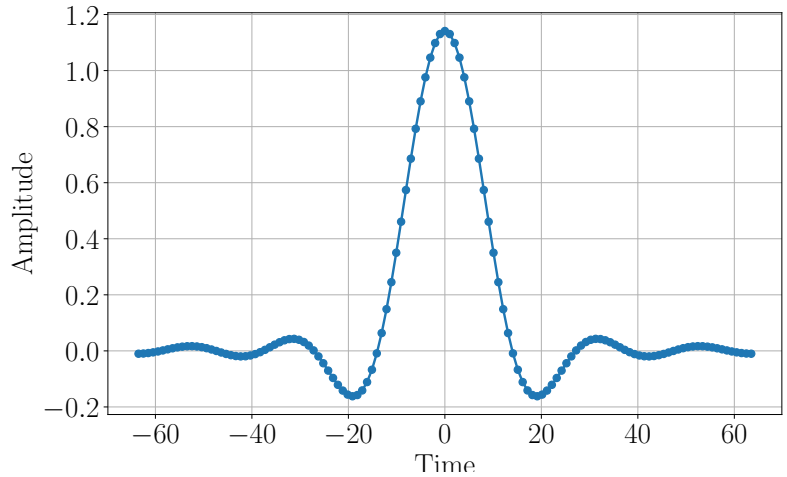


Figure 4.1: Root-Raised-Cosine Pulse Shaping Function

In this model, a_k denotes the QPSK symbols, which are mapped from the input bits. Initially, the bits are mapped to QPSK symbols and subsequently passed through the pulse shaping filter to produce the signal $s[n]$. The interference signal b originates from one of four datasets, representing different types of interference. The exact characteristics of the interference signals are generally unknown, as the task requires blind interference mitigation. The Signal-to-Interference-plus-Noise Ratio (SINR) ranges from -30 dB to 0 dB in our evaluations. The received signal y is modeled as a combination of the QPSK signal of interest and the interference:

$$y[n] = s[n] + b[n], \quad n = 0, 1, \ldots, L-1. \tag{4.2}$$

The objective is to extract the signal of interest s from the received mixture y, despite the presence of unknown interference b. This interference signal b originates from one of four datasets used in the competition, and as such, the exact characteristics of the interference were unknown even to us. This signal model provides the foundation for the signal processing and detection algorithms developed to achieve effective separation and mitigation of interference, ensuring robust communication in challenging environments.

4.2.2 Error Metric and Loss Function

To evaluate the performance of our system, we use the Mean Squared Error (MSE) between the true signal $s[n]$ and its estimated version $\hat{s}[n]$. The MSE is defined as:

$$\text{MSE} = \frac{1}{L}\sum_{n=0}^{L-1}(s[n] - \hat{s}[n])^2. \tag{4.3}$$

Additionally, we introduce an MSE score based on the truncated negative logarithmic value of the MSE. This truncation is applied because MSE values better than a certain threshold do not significantly enhance the ultimate goal of interference cancellation, which is bit decoding. By capping the MSE score, we prevent the model from focusing on overly precise sample reconstructions in some examples and poor reconstruction in other that may lead to high MSE score and poor average Bit Error Rate (BER). Instead, the model is encouraged to achieve uniformly good reconstructions across all samples, improving the overall BER. In our work, we set the threshold at -50 dB [4].

Although we use this MSE score for validation and testing, during training, we employ a smoothed version of this loss function to facilitate stable and continuous training. The smoothing function used is a sigmoid-type function, which helps to avoid discontinuities in the loss landscape. This approach ensures that our model not only achieves low MSE but also maintains a reconstruction quality sufficient to enhance BER, thereby achieving effective interference mitigation and robust communication. In the sequel, we solely focus on the MSE score, and a BER analysis that includes bit decoding is left open for future research and studies.

4.3 Methodology

This research introduces a U-Net model tailored for the task of RF signal separation [5], with a focus on OFDM signals. The architecture of our model is a U-Net, comprising six encoder layers and five decoder layers.

4.3.1 Model Architecture

In the OFDM signal structure, each symbol is comprised of 80 samples, of which the first 16 are the Cyclic Prefix (CP), and the remaining 64 correspond to the subcarriers. Out of these 64 subcarriers, 56 carry nonzero symbols [3]. Our model applies a short-time Fourier transform (STFT) with an FFT size of 64 and a hop length of 80, ensuring each FFT computation captures only the 64 subcarrier samples and skips the 16-sample CP, aligning precisely with each OFDM symbol. Conversely, at the output of the model, the inverse STFT (iSTFT) is applied to reconstruct the time-domain signal. The kernel size for each layer is consistently

set to 3, and group normalization is implemented across all layers with a size of 8 to enhance the model's ability to generalize. Our U-Net model, Fig. 1, employs a configuration of varying encoder strides [1, 2, 2, 2, 2, 2] and encoder filters [192, 256, 384, 512, 512, 512]. Additionally, we utilize residual connection-based convolution blocks in our network architecture, which effectively prevent the vanishing gradient problem while adding depth and complexity.

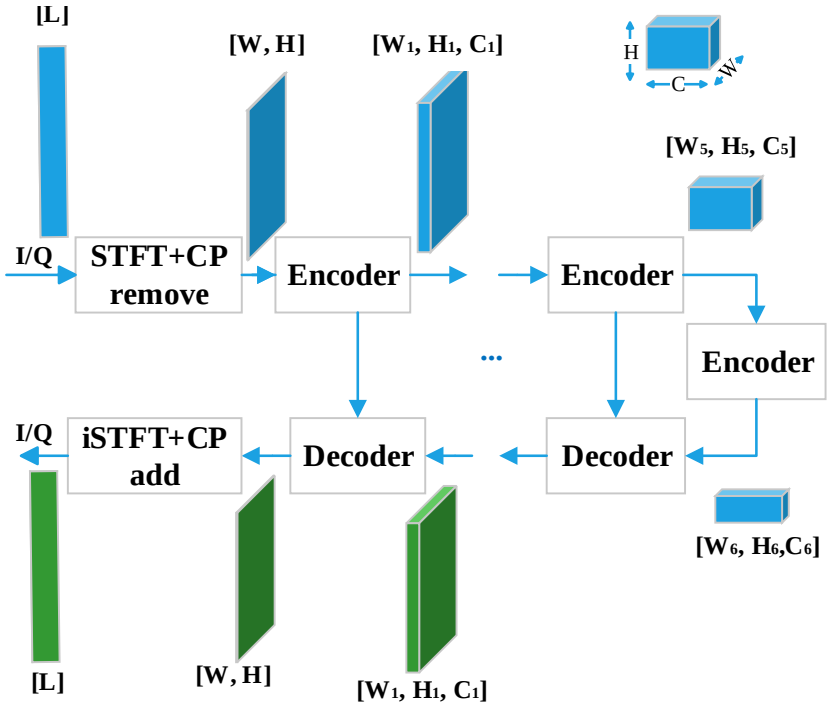


Figure 4.2: Schematic of the interference cancellation model. The I/Q baseband RF signal is first transformed using the Short-Time Fourier Transform (STFT) and Cyclic Prefix (CP) removal to obtain a time-frequency representation of dimensions [W, H]. This representation is processed by encoder and decoder networks and then reconstructed via inverse STFT (iSTFT) with CP addition.

4.3.2 Data Handling and Preprocessing

The dataset utilized follows the guidelines set by the challenge organizers [3]. To augment the dataset, we apply transformations to both the phase and amplitude of the interference signal. During training, we process the complex-valued input samples into real values, training them with a batch size of 2. The data is formatted

into two separate channels representing the real and imaginary parts, conforming to a structure of $[batch, channel, length]$, where $batch = 2$, $channel = 2$, and $length = 40960$. We employ 5-fold cross-validation to ensure distinct separation of training and validation sets.

4.3.3 Model Parameters and Configuration

The model is trained using a truncated mean squared error loss, aligning with the challenge's scoring criteria [3]. The optimization is carried out using the Adam optimizer, with an initial learning rate of 0.0002 and a cosine annealing scheduler. This setup is selected to achieve a balance between rapid convergence and avoiding local minima. For testing, we utilize separate frames that were not included in the training and validation datasets. This approach ensures the robustness and generalizability of the model's performance.

4.4 Results

Table 4.1 demonstrates the performance of the proposed U-Net model as compared to no mitigation, LMMSE, as well as U-Net and WaveNet baselines. Utilizing a 2D U-Net architecture for OFDM signal processing, our model achieved average MSE values: -19.31 dB for EMI, -13.32 dB for Comm2, -4.84 dB for Comm3, and -14.39 dB for Comm5G scenarios. These figures notably surpass those of challenge baseline methods, such as Baseline WaveNet and Baseline U-Net, across all interference types. While our 2D U-Net model excelled in the OFDM signal-of-interest (SOI), a different approach was adopted for the QPSK SOI, where a 1D U-Net model was utilized. This distinction is crucial, as it highlights the model's adaptability to different signal characteristics. The superior performance of the 2D model in OFDM scenarios can be attributed to its design. This tailored approach to handling the unique aspects of OFDM signal structures, such as dealing efficiently with the CP, likely contributes to its outstanding performance in these contexts.

4.5 Discussion and Conclusion

In this work, we presented a U-Net-based architecture tailored for the separation of co-channel interference signals in the time-frequency domain, with a specific focus on OFDM signals. By leveraging the structural properties of OFDM, such as the cyclic prefix, and employing a carefully designed 2D U-Net model, we achieved significant improvements in signal separation performance. Our results, with an average MSE enhancement of 63% over baseline methods across four interference types (EMI, Comm2, Comm3, and Comm5G), underscore the efficacy of integrating domain-specific knowledge into deep learning frameworks. This approach not only

Table 4.1: Performance comparison in terms of average MSE (in dB)/BER score for SINRs ranging from -30 to 0 dB across various interference types. Our proposed method improves MSE (linear) by 76.93%, 78.67%, 43.77%, 55.84% for EMI, Comm2, Comm3, and Comm5G, respectively.

Method	EMI	Com2	Com3	Com5G
No Mitigation	15/0	15/0	15/0	15/0
LMMSE	-3.3/0	-2.0/0	-1.9/0	-5.4/0
Base U-Net	-9.2/1.6	-5.4/0.8	-1.8/0	-8.4/0
Base WaveNet	-12.9/1.6	-6.6/0.8	-2.3/0	-10.8/1.6
Proposed Model	**-19.3/2.7**	**-13.3**/0.8	**-4.8**/0	**-14.4**/1.6

outperforms traditional methods like LMMSE but also surpasses general-purpose deep learning baselines such as WaveNet and standard U-Net configurations.

However, future work could explore the model's efficiency and extend its applicability to deployment in emerging communication standards like 6G. Additionally, incorporating real-time processing capabilities could further enhance its practical utility in dynamic wireless environments. This research lays a strong foundation for advancing data-driven solutions in interference mitigation, contributing to more reliable and efficient communication systems.

References

[1] S. Lutati, E. Nachmani, and L. Wolf. *Separate and diffuse: Using a pre-trained diffusion model for improving source separation*. arXiv preprint arXiv:2301.10752, 2023.

[2] G. C. Lee, A. Weiss, A. Lancho, Y. Polyanskiy, and G. W. Wornell. *On Neural Architectures for Deep Learning-Based Source Separation of Co-Channel OFDM Signals*. In ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, June 2023. Available from: http://dx.doi.org/10.1109/ICASSP49357.2023.10096702, doi:10.1109/icassp49357.2023.10096702.

[3] T. Jayashankar, B. Kurien, A. Lancho, G. C. Lee, Y. Polyanskiy, A. Weiss, and G. Wornell. *The Data-Driven Radio Frequency Signal Separation Challenge*. In To appear in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), April 2024.

[4] T. Jayashankar, B. Kurien, A. Lancho, G. C. Lee, Y. Polyanskiy, A. Weiss, and G. W. Wornell. *The Data-Driven Radio Frequency Signal Separation Challenge*. In Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 2024.

[5] A. Lancho, A. Weiss, G. C. Lee, J. Tang, Y. Bu, Y. Polyanskiy, and G. W. Wornell. *Data-driven blind synchronization and interference rejection for digital communication signals*. In GLOBECOM 2022-2022 IEEE Global Communications Conference, pages 2296–2302. IEEE, 2022.

5

Blind Co-channel Interference Cancellation Using Fast Fourier Convolutions

Efficiently canceling co-channel interference in wireless systems remains a formidable challenge, particularly when waveforms exhibit long-range dependencies. Many existing convolutional network solutions rely on either large kernels or deep architectures to capture these dependencies, often at the cost of higher computational overhead. This chapter introduces a streamlined UNet model that leverages fast Fourier convolution blocks and includes a long short-term memory (LSTM) component in its bottleneck, thus reducing complexity without compromising performance.

Addressing long-range dependencies in blind co- channel interference waveforms typically requires convolutional networks with large kernels or significant depth, which are resource-intensive. This chapter presents a streamlined UNet architecture integrated with fast Fourier convolution blocks and a long short-term memory in the bottleneck, designed to efficiently capture these dependencies. By leveraging the Fourier domain for global feature processing, our architecture reduces the model's complexity without compromising performance. Compared to the leading benchmark model (a deep UNet), our approach yields a 26.5multiply-accumulate operations and the number of model param- eters by 76.8cant enhancement in both accuracy and efficiency for interference cancellation in constrained computational environments.

Building on traditional convolutional architectures, the proposed design streamlines

computational demands by transforming waveforms into the Fourier domain. This enables global feature extraction with fewer operations compared to purely time-domain convolutions. In addition, placing an LSTM in the bottleneck layer helps capture extended temporal correlations, further improving the effectiveness of interference cancellation.
Throughout this chapter, we detail the architectural components that yield substantial performance gains over deep UNet benchmarks. We also discuss how reductions in multiply-accumulate operations (MACs) and model parameters translate to faster inference and lower power consumption—qualities that are especially relevant for edge or resource-constrained scenarios. The ensuing sections cover the motivations behind fast Fourier convolutions, the network's training paradigm, and comprehensive performance evaluations demonstrating clear advantages in both accuracy and efficiency.

⋆ ⋆ ⋆

This Chapter is adapted from:

Mostafa Naseri, Eli De Poorter, Ingrid Moerman, H Vincent Poor, Adnan Shahid
Blind Co-Channel Interference Cancellation Using Fast Fourier Convolutions

Published in Vehicular Technology Conference, 2024.

This chapter targets C2, C3, and C4.

5.1 Introduction

Source separation is integral to signal processing, underpinning advancements in diverse applications such as audio enhancement and wireless communication. With the escalating utilization of the radio frequency spectrum, wireless networks encounter a growing challenge in managing interference. This congestion necessitates advanced signal isolation and cancellation techniques to ensure reliable communication.

Blind interference cancellation—where interfering signals are removed without prior knowledge of their characteristics— is crucial in scenarios where interference characteristics are unknown. It enables wireless systems to adapt to a variety of interference patterns, including those with long-range dependencies that are typically challenging to decode. This work introduces a UNet architecture integrated with fast Fourier convolutions (FFC) [1], featuring an LSTM in the bottleneck. The

research by Shchekotov et al. [2] showed the efficacy of FFC operators in equipping neural networks with expansive receptive fields in initial layers, a feature beneficial for handling periodic structures in audio signals. This aspect of FFCs is particularly relevant to our work, considering the periodic nature of some radio interference sources, as exemplified in the EMISignal1 dataset, which will be discussed later in this paper. This configuration yields significant improvements in mean square error (MSE) and reduces computational and memory demands. This approach enables advanced signal processing in resource-constrained devices and improves bandwidth efficiency for model weight exchange in cooperative learning, due to smaller model size.

5.2 Methodology

5.2.1 System and Data Model

In this study, we adopt the system model described in sec. 4.2, with the exception that the interference type is unknown in this context. We consider a signal model where the received signal $y = s + b \in \mathbb{C}^L$ is a complex-valued mixture of length L. This mixture comprises the signal of interest (SOI) s and an interference signal b, where s and b overlap in both time and frequency. The SOI, s, is modeled as a QPSK signal shaped with a root raised cosine waveform following the configurations of the data-driven signal separation in radio spectrum: ICASSP 2024 [3] (roll-off factor of 0.5). The interference signal b is derived from one of four datasets provided by the challenge, representing different types of interference:

EMISignal1: Electromagnetic interference from a man-made source. **CommSignal2** and **CommSignal3:** Different digital communication signals from commercial wireless devices. **CommSignal5G1:** A 5G-compliant waveform.

Unlike the data-driven signal separation in radio spectrum challenge which assumes prior knowledge of the interference type, our approach operates under the assumption of no prior knowledge regarding the nature of the interference. Interference signals b are sampled randomly from these datasets and augmented with a random amplitude and phase shift to represent varying SINR levels and phase variations. This augmentation strategy is employed during training to enhance the generalizability of the model. The SINR values used in our study range from -30 dB to 0 dB.

5.2.2 Model Architecture

Our model adopts a UNet framework augmented with FFC blocks in the encoder, a bidirectional LSTM in the bottleneck for sequential data processing, and a nuanced double convolution decoder for signal reconstruction.

The encoder employs FFCRes blocks, fig. 5.1, each consisting of two paths: one adjusts the input's temporal dimension through a matching convolution, preparing it for residual addition; the other processes the input through two sequential FFCs; the output of these paths is then combined.

The model's bottleneck features a bidirectional LSTM layer to effectively handle the temporal dependencies crucial for signal separation, while also avoiding the gradient issues common in traditional RNNs. We also implement a transformer encoder layer as bottleneck (FFC-TF) to show the effectiveness of our proposed FFC-LSTM architecture.

The decoder upsamples the signal via a strided convolutional transpose layer, applies group normalization and LeakyReLU, and then combines it with encoder features through skip connections via another convolutional layer.

Training was executed with a learning rate of 2×10^{-4} and a batch size of 2 over 10^6 steps using the Adam optimizer. For stability, a soft MSE loss with truncation at -50 dB was used for training, shifting to hard truncation for evaluations. The dataset was split into 80% for training and 20% for validation, reserving 50 frames for final assessment; a cosine annealing scheduler was employed to optimize learning. The robustness of our architecture was validated through 50 independent trials, each with 50 different random selection of windows from the untouched test dataset. These trials spanned 11 distinct SINR values in the range of $[-30, 0]$ dB, ensuring a comprehensive assessment of performance across varying conditions.

5.3 Quantitative Results

In our evaluation, the model was benchmarked against established architectures: Vanilla UNet, DeepMind's WaveNet (referenced in [3]), and ConvTasNet, using their standard configurations. The primary metric for performance assessment was MSE, with a secondary emphasis on computational efficiency, measured by multiply-accumulate operations (MACs).

Our findings, fig. 5.2, indicate that the FFC encoder outperformed the vanilla UNets, achieving lower MSE values at reduced MACs. Specifically, the FFC-equipped bi-LSTM model surpassed the UNet (with encoder filters $[64,128,256]$), UNet (v1) ($[64,128,256,512,512,512]$) and UNet (v2) ($[64,128,256,512,1024,2048]$) in terms of average MSE. The plot illustrates the trade-off between computational complexity (MACs) and performance, highlighting the efficiency of our proposed method. The analysis of computational complexity also revealed that the number of parameters in a model does not directly correlate with computational complexity, as evidenced by models with larger parameter counts not necessarily incurring more MACs.

In summary, the FFC encoder equipped with a bi-LSTM architecture presents a significant improvement in the domain of blind co-channel interference cancel-

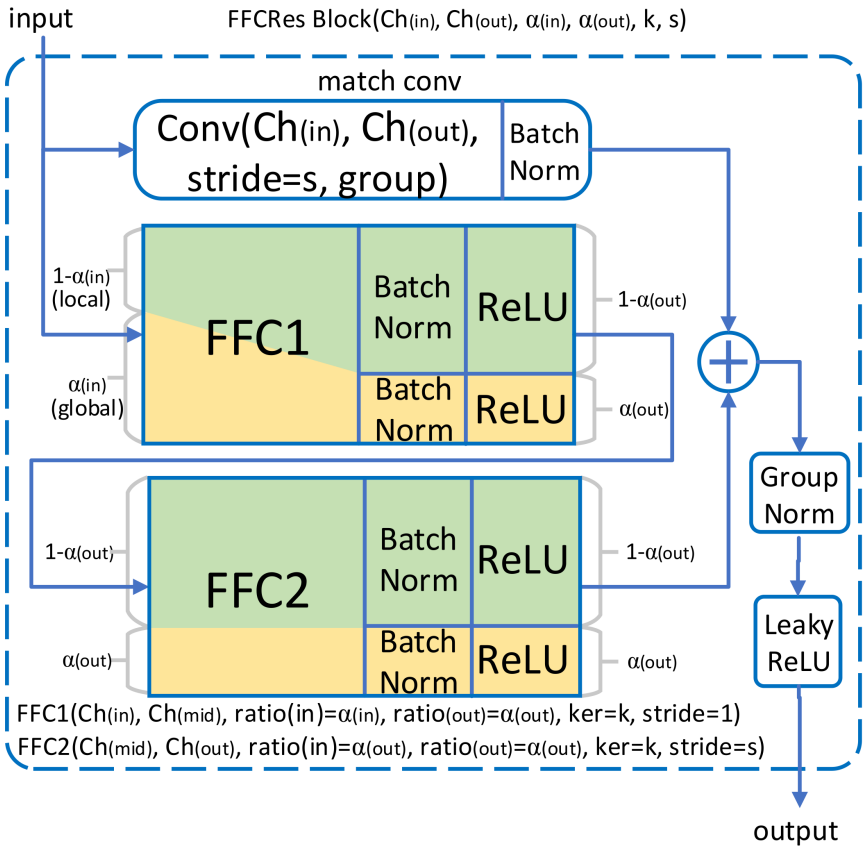


Figure 5.1: The Fast Fourier Convolution (FFC) residual block used in the UNet encoder. This block includes a matching convolution to align input and output dimensions. FFC layers are configured with filter sizes $[64, 128, 256]$*, a global ratio of* $\alpha = 0.5$*, kernel size* $k = 3$*, and stride* $s = 2$*.*

lation. This is substantiated by its superior performance in MSE reduction and computational as well as memory efficiency, as demonstrated in the comparison with established models.

5.4 Discussion and Conclusion

This study introduces a U-Net architecture augmented with FFC blocks and a bidirectional LSTM in the bottleneck to address blind co-channel interference cancellation. The results indicate that the integration of FFC for global feature processing and LSTM for temporal modeling effectively captures long-range de-

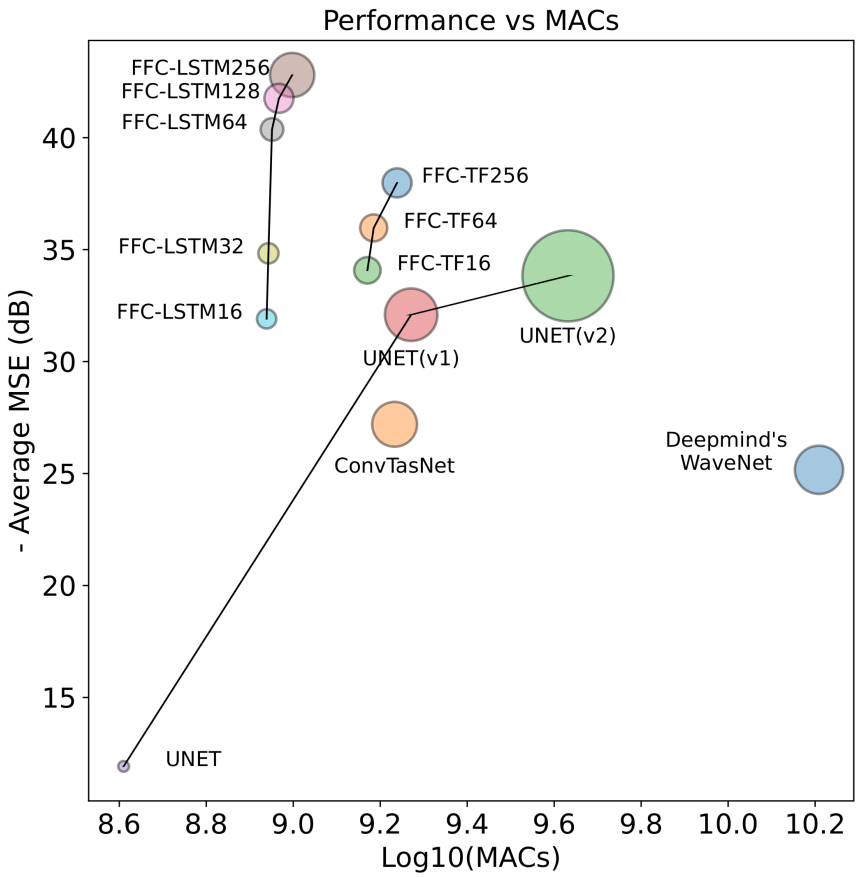


Figure 5.2: Scatter plot of average (over SINRs) MSE vs. computational complexity. Circle sizes denote parameter counts. FFC-LSTM models are marked by LSTM hidden state sizes, while FFC-TF models show transformer feedforward dimensions. The proposed FFC encoder with bi-LSTM achieves lower MSE at reduced MACs compared to baselines.

pendencies in interference waveforms. Quantitative evaluation shows a reduction in mean square error alongside a decrease in computational complexity and model parameters compared to a deep U-Net baseline, suggesting a viable trade-off between performance and efficiency.

The model's ability to handle diverse interference types without prior knowledge of their characteristics highlights its potential adaptability. This is particularly relevant given the increasing spectral congestion in wireless networks, where efficient signal separation remains a critical challenge. The reduced resource demands also point to its suitability for constrained computational environments, though practical limitations, such as training stability and dataset variability, warrant

further exploration.

Looking ahead, future efforts could focus on real-time implementation to enable dynamic interference cancellation in operational settings. Additionally, advancing towards the actual deployment of interference cancellation (IIC) models in real-world wireless systems could bridge the gap between theoretical performance and practical utility. Investigating these directions may enhance the applicability of this approach, contributing to more robust signal processing solutions in modern communication networks.

References

[1] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky. *Resolution-robust large mask inpainting with fourier convolutions*. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 2149–2159, 2022.

[2] I. Shchekotov, P. Andreev, O. Ivanov, A. Alanov, and D. Vetrov. *FFC-SE: Fast fourier convolution for speech enhancement*. arXiv preprint arXiv:2204.03042, 2022.

[3] A. Lancho, A. Weiss, G. C. Lee, J. Tang, Y. Bu, Y. Polyanskiy, and G. W. Wornell. *Data-driven blind synchronization and interference rejection for digital communication signals*. In GLOBECOM 2022-2022 IEEE Global Communications Conference, pages 2296–2302. IEEE, 2022.

6

High-Throughput Interference Cancellation

As wireless networks grow in complexity and data rates escalate, the need for efficient co-channel interference (CCI) cancellation techniques becomes ever more pressing, particularly for edge devices with stringent resource constraints. While blind CCI solutions using machine learning (ML) have demonstrated promise, maintaining high throughput with limited computational overhead remains a key challenge. In this chapter, we introduce an optimized U-Net architecture equipped with depthwise separable convolutions, quantization, and pruning strategies to strike an optimal balance between interference cancellation performance and computational efficiency.

Co-channel interference cancellation (CCI) is the process used to reduce interference from other signals using the same frequency channel, thereby enhancing the performance of wireless communication systems. An improvement to this approach is blind CCI, which reduces interference without relying on prior knowledge of the interfering signal characteristics. Recent work suggested using machine learning (ML) models for this purpose, but high-throughput ML solutions are still lacking, especially for edge devices with limited resources. This work explores the adaptation of U-Net Convolutional Neural Network models for high-throughput blind source separation. Our approach is established on architectural modifications, notably through quantization and the incorporation of depthwise separable convolution, to achieve a balance between computational efficiency and performance. Our results demonstrate that the proposed models achieve superior MSE scores

when removing unknown interference sources from the signals while maintaining significantly lower computational complexity compared to baseline models. One of our proposed models is deeper and fully convolutional, while the other is shallower with a convolutional structure incorporating an LSTM. Depthwise separable convolution and quantization further reduce the memory footprint and computational demands, albeit with some performance trade-offs. Specifically, applying depthwise separable convolutions to the model with the LSTM results in only a 0.72% degradation in MSE score while reducing MACs by 58.66%. For the fully convolutional model, we observe a 0.63% improvement in MSE score with even 61.10% fewer MACs. Additionally, the models exhibit excellent scalability on GPUs, with the fully convolutional model achieving the highest symbol rates (up to 800×10^3 *symbol per second) at larger batch sizes. Overall, our findings underscore the feasibility of using optimized machine-learning models for interference cancellation in devices with limited resources.*

In contrast to conventional CCI techniques that rely on known interference patterns, our approach focuses on blind source separation, thereby accommodating a wide range of interference types without explicit prior information. We demonstrate how depthwise separable convolutions reduce the parameter count and computational overhead, and how quantization leads to further efficiency gains with minimal performance sacrifice. Pruning strategies are also discussed, along with their effect on inference speed and model size.

The chapter is organized to highlight design considerations for resource-constrained hardware, demonstrating how both a deeper, fully convolutional model and a shallower, LSTM-enhanced model can achieve strong MSE performance while preserving throughput. We further showcase the impact of batch size scaling on GPU-based inference, revealing the potential for extremely high symbol rates. Overall, our results contribute to the broader understanding of how U-Net-based architectures can be optimized for low-latency, energy-efficient interference cancellation—a capability crucial to next-generation wireless systems operating at the network edge.

⋆ ⋆ ⋆

This Chapter is adapted from:

Mostafa Naseri, Eli De Poorter, Ingrid Moerman, H. Vincent Poor, Adnan Shahid
High-Throughput Adaptive Co-Channel Interference Cancellation for Edge Devices Using Depthwise Separable Convolutions, Quantization, and Pruning

Published in IEEE Open Journal of the Communications Society, 2024.

This chapter targets C2 and C3.

6.1 Introduction

Signal separation, an integral technique in signal processing, has a broad spectrum of applications across diverse fields. In audio signal processing, it plays a critical role in noise reduction for speech and music recordings, separating vocals from instrumentals, and enhancing clarity in hearing aids [1]. This technology not only improves the quality of audio content but also aids in accessible communication. In the realm of medical imaging, signal separation techniques are pivotal for artifact removal in MRI and CT scans, contributing to clearer and more accurate diagnostic images [2]. It is also instrumental in analyzing complex signals in electroencephalography (EEG) and magnetoencephalography (MEG), enhancing our understanding of neural activities. Financial time series analysis also benefits from signal separation. It aids in filtering out market noise, allowing analysts to identify informative trends and underlying factors that influence asset prices, thus facilitating more informed investment decisions [3]. In the field of astronomy and space exploration, signal separation is key in filtering cosmic noise in radio telescope data and is integral in the signal processing for satellite communication and operations in the deep space network [4]. Furthermore, in video processing, signal separation techniques are used for noise removal and feature enhancement in surveillance footage and for separating and reconstructing overlaid images or videos, providing clearer visual data for various applications.

6.1.1 Importance of Co-Channel Interference Cancellation

In the evolving landscape of wireless communications, Co-channel Interference (CCI) emerges as an interesting challenge, especially due to the increasing demand for spectral efficiency and the need for high-quality signal transmission. Primarily caused by spectral reuse—a practice where the same frequency bands are deployed across different transmitters—CCI significantly compromises the Signal-to-Interference-plus-Noise Ratio (SINR), a crucial determinant of communication reliability and quality.

As wireless technologies spread across various platforms, from satellites to terrestrial vehicles, and as edge devices become widespread, the issue of CCI grows more severe. These devices frequently encounter a stream of composite signals, composed of both desired and undesired transmissions, which severely degrade the SINR. Effective interference cancellation mechanisms are therefore essential to process and extract the intended signals from this interference, enhancing signal clarity and overall communication fidelity. Furthermore, the concept of underlay

communication, particularly relevant in the context of next-generation 5G networks, highlights another critical application of interference cancellation [5–7]. Underlay strategies allow secondary users—such as unmanned aerial systems and remote sensors—to operate within the same spectral and temporal resources as primary users. This method not only facilitates innovative uses of existing spectral resources but also significantly boosts spectrum efficiency, showcasing the profound relevance of advanced CCI cancellation techniques in modern telecommunication systems.

The importance of addressing CCI in modern communication systems is driven by several key technological advancements and emerging market demands. Notably, the growth of safety-critical systems [8], such as those used in automotive and healthcare applications, necessitates highly reliable communications even in environments disturbed by strong interference. This has encouraged interest in technologies like successive interference cancellation, which can dramatically enhance the quality and reliability of signal reception in congested scenarios [7]. Furthermore, the expansion of Internet of things (IoT) devices, which often operate in dense networks with limited spectral resources, calls for more sophisticated interference management techniques. Vehicle-to-Anything communication (V2X) communications, crucial for the safety of autonomous and connected vehicles, also underscore the need for robust interference cancellation mechanisms to ensure uninterrupted communication in dynamic environments.

In addition, there is growing interest in exploiting unlicensed spectrum and cognitive radio networks to enhance spectrum efficiency. These technologies rely heavily on advanced interference cancellation to prevent disruption of existing services while supporting new wireless applications. Satellite communications, another area experiencing rapid growth due to global connectivity initiatives, face unique challenges related to CCI due to the overlapping signals from multiple sources. Together, these factors make the study of CCI cancellation techniques not only relevant but essential for the next generation of wireless systems, driving forward innovations that will shape the future of global communications. As the complexity and density of wireless networks continue to grow, traditional signal processing approaches struggle to keep up with the dynamic and unpredictable nature of modern interference patterns. These methods often rely on predefined models and assumptions that may not accurately capture real-world conditions, leading to suboptimal performance in practical scenarios. While the integration of machine learning models into interference cancellation strategies marks a progressive step toward smarter communication systems [9], it also unveils significant challenges. Specifically, the computational limitations of nodes within modern networks pose a considerable bottleneck. Most existing machine learning algorithms for CCI cancellation demand high computational resources, which can be impractical in real-time applications where quick processing is critical [10, 11]. This paper addresses these crucial gaps by proposing and investigating efficient neural network architectures

and techniques that are tailored for real-time interference cancellation without compromising performance. This approach seeks to bridge the gap between the theoretical potential of machine learning models and their practical implementation in communication networks disurbed by CCI.

6.1.2 Related Work

6.1.2.1 Background on Blind Co-Channel Interference

The history of CCI management is marked by continuous evolution, beginning with simple linear filters in analog signal processing. These filters, albeit basic, laid the groundwork for more sophisticated non-linear and predictive methods [12]. With digital signal processing, adaptive filters emerged, dynamically adjusting to signal environments and outperforming their analog predecessors. The spread of spectrum technology, including frequency-hopping and CDMA, offered robust solutions to reduce CCI by dispersing the signal across a broader bandwidth. The implementation of Multiple Input Multiple Output (MIMO) technology marked a significant leap. Utilizing multiple antennas at both transmission and reception points, MIMO systems harnessed spatial diversity and multiplexing to enhance capacity and mitigate interference [13].

The advent of machine learning, especially deep learning techniques like convolutional neural networks (CNNs) and recurrent neural networks (RNNs), promises a new era in CCI management. These networks excel in modeling complex interference patterns and separating sources in densely populated signal environments. However, most machine learning (ML)-based CCI studies focus on known interference types, as highlighted in [14]. Notably, models like WaveNet, though efficient in handling known interference [14], exhibit significant performance degradation in blind or unknown interference scenarios [15].

Furthermore, in the realistic conditions of wireless networks, where we frequently encounter interference with unknown characteristics, the ability to cope effectively becomes essential. This necessity leads us to explore the realm of blind CCI cancellation. In such a scenario, our methods must proficiently address interference without prior insight into its specific properties.

6.1.2.2 Challenges in Real-Time Interference Cancellation

Achieving real-time interference cancellation in wireless communications encompasses several key aspects: inference time, throughput, and consistency. Inference time, particularly, refers to the time taken by a model to process input and generate output, exclusive of delays caused by data acquisition, pre/post-processing, and other systemic components. In this study, we specifically focus on the inference time as a crucial factor in determining real-time capability.

The primary challenges in achieving real-time interference cancellation are multi-fold. First and foremost is the computational complexity of the models. Advanced algorithms capable of effective interference cancellation often require significant computational resources, which can lead to increased processing time. In addition to processing power, memory constraints in edge devices limit the size and complexity of deployable models, thus posing a challenge to maintaining high performance under limited resources. Another critical challenge is power consumption. Edge devices, designed for efficiency and prolonged operation, necessitate models that are not only fast but also energy-efficient. This becomes a delicate balance, as increased model complexity and processing speed often translate into higher power consumption.

Previous approaches to interference cancellation have predominantly focused on performance metrics such as accuracy, often neglecting the aspect of inference time. For instance, we will show that models like WaveNet, despite having moderate Floating Point Operations Per Second (FLOPS), exhibit notably low processing speeds, likely due to the implementation specifics of their dilated convolutions. This disparity highlights the necessity of considering inference speed.

The importance of real-time processing in CCI cancellation cannot be overstated. In many practical scenarios, especially in wireless communication, the delay sensitivity of interference cancellation and bit decoding makes the feasibility of machine learning models dependent on their real-time performance capabilities. Delays can significantly undermine the practicality of deploying these models in real-world scenarios. By designing a high-speed implementation, we aim to reduce the inference delay, thereby making real-time interference cancellation feasible on edge devices. This directly tackles the challenges of computational complexity, memory constraints, and power efficiency, paving the way for practical and efficient machine learning-based CCI cancellation in resource-limited environments.

6.1.3 Objective of the Study

The primary objective of this study is to advance the implementation of real-time, CNN-based models for CCI cancellation. This research aims to bridge the gap between high computational demand and real-time processing requirements, especially in scenarios involving edge devices and mobile communications. To achieve this, we thoroughly examine the relationship between inference time, computational complexity, and memory footprint of these models.

Specifically, our study involves a detailed exploration of both full-precision and quantized (qint8) models. The choice of quantization aligns with our goal of enabling efficient CCI cancellation on a broader spectrum of devices, including those with limited computational resources.

In terms of performance metrics, we employ symbols per second decoded by the

model as a measure of inference speed, Multiply-Accumulate Operations (MACs) for computational complexity, and the number of model parameters to assess memory footprint. These metrics collectively offer a comprehensive view of the model's efficiency and practicality in real-world applications.

The expected outcome of this research is to enable higher quality communication for mobile and edge devices through improved signal quality post-CCI cancellation. This improvement has the potential to reduce power consumption, especially in systems with battery constraints, making it particularly relevant for Wireless Sensor Networks (WSNs) as well as other mobile communications scenarios. Additionally, the possibility of integrating CCI cancellation with decoding processes offers a promising avenue for enhancing Radio-Frequency Integrated Circuits (RFICs), contributing to more energy-efficient and robust communication systems.

Mainly, this study has broader implications for energy consumption in mobile and edge devices. By optimizing CNN models for CCI cancellation, we can substantially lower the energy requirements of these devices, making a significant contribution to the sustainability and efficiency of wireless communication networks globally.

6.1.4 Main Contributions

The main contributions of this paper are:

- Design of a set of convolutional neural networks for CCI cancellation that outperforms the baseline models from the scientific literature.
- Improvement of the proposed model architectures through optimization techniques for better performance.
- Development of a fully convolutional architecture that is independent of input size.
- Comprehensive evaluation of the models on both CPU and GPU, testing the scalability on GPU.

The remainder of this paper is structured as follows. In sec. 4.2, we present the system model and discuss the error metric and loss function used. Sec. 6.2 outlines our methodology, detailing the model architecture, including the U-Net CNN model, depthwise convolution, and baseline models, as well as the quantization techniques employed. We also describe the dataset, data preparation, training process, and evaluation metrics. Sec. 6.3 provides the results, highlighting model performance, complexity, inference rate, comparative analysis, GPU parallelizability, and the impact of pruning. In Sec. 6.4, we analyze the results and discuss the implications

of our findings, along with future work directions. Finally, sec. 6.5 concludes the paper.

6.1.5 System and Data Model

In this study, we adopt the system model described in sec. 4.2, with the exception that the interference type is unknown in this context.

6.2 Methodology

In this study, we adopt a structured approach to address the challenge of CCI cancellation by proposing two neural network architectures. Our methodology centers around the utilization of the U-Net architecture, renowned for its effectiveness in various signal and image processing tasks [16, 17]. We propose two distinct variants of the U-Net model tailored specifically for CCI cancellation: a shallow U-Net characterized by small numbers of filters at each layer and an long short-term memory (LSTM) at the bottleneck (M1), and a deeper U-Net equipped with a larger number of filters (M2).

To enhance the performance and efficiency of these models, we introduce two significant modifications: the integration of depthwise convolutions and the application of quantization techniques. Depthwise convolutions are employed to reduce the computational demand while maintaining model effectiveness, and quantization is applied to further enhance the operational efficiency, making the models more suitable for deployment in real-time communication systems.

Each modified model's performance is systematically compared against baseline models, which will be introduced and described in subsequent sections. This comparative analysis aims to highlight the advantages of our proposed solutions in terms of both efficiency and effectiveness in CCI cancellation.

6.2.1 Model Architectures

The architecture of the proposed models is guided by a set of design principles aimed at optimizing performance for CCI cancellation while maintaining computational efficiency and robustness. The development process involved a meticulous hyperparameter search to achieve an optimal balance between model complexity and performance. Starting with highly complex models, the number of parameters was progressively reduced. This parameter tuning was carefully managed to prevent any significant drop in performance, ensuring that both variants of the U-Net model—referred to as M1 and M2—are lightweight yet effective.

Efficiency and Scalability: Both models were designed to be scalable and adaptable to different data types and input sizes. M2, being fully convolutional,

showcases exceptional flexibility, allowing it to handle various input lengths during inference and to be retrained with different datasets without structural modifications. This design choice provides substantial advantages in terms of deployment in diverse operational environments. In contrast, M1 can practically handle different input lengths, but unlike M2, significant deviations from the input size used in training can degrade its performance (due to stateful nature of LSTM).

Robustness: To enhance the models' robustness, data augmentation techniques were employed extensively. These techniques, which will be detailed in subsec. 6.2.3 section, help the model generalize better across different interference scenarios. Unlike the approach taken in [14], where separate models were trained for each type of interference, a single model in this study is trained across all available data types, which enhances its applicability and efficiency in real-world settings.

Temporal and Spatial Feature Extraction: The M1 model incorporates an LSTM layer at its bottleneck, focusing on capturing temporal dependencies and long-range dynamics within the signal. This is particularly beneficial for scenarios where the interference exhibits time-dependent characteristics. In contrast, M2 prioritizes spatial feature extraction through a network structure that is deeper and has a larger number of filters. This approach is designed to exploit the intricate spatial patterns of signals, making it suitable for environments with complex interference structures.

Two critical modifications, depthwise convolutions and quantization, have been implemented to optimize the performance and efficiency of the proposed neural network architectures.

Depthwise Convolutions: The introduction of depthwise convolutions in the models marks a significant advancement in reducing computational complexity [18]. By performing convolutions in a separable manner—where the spatial and depth (channel) convolutions are decoupled—the number of trainable parameters and the computational demand, measured in MACs, are substantially lowered. Although the actual inference speed gains from depthwise convolutions can vary depending on the platform, this modification inherently decreases the computational load, making the models more adaptable to a variety of deployment scenarios, including those with limited processing capabilities.

Quantization: The application of quantization techniques aims to further reduce the model size and computational requirements. By lowering the precision of the numerical values used in the model, quantization not only shrinks the overall memory footprint but also enhances the feasibility of implementing these models on processors with lower precision capabilities. This adjustment is crucial for deploying advanced neural network architectures in real-time communication systems where hardware efficiency is paramount.

6.2.1.1 U-Net CNN Model

The first U-Net Model, referred to as M1, is structured to effectively address the challenges of CCI cancellation through a series of encoding and decoding layers, supplemented by a specialized LSTM layer at its bottleneck.

M1 Encoder: The encoder part of M1 consists of three layers, each equipped with a double convolution mechanism to enhance feature extraction. The number of filters in these layers scales up through [64, 128, 256] with corresponding strides [1, 2, 2], allowing the model to capture features from the input signal at various resolutions. in all convolutional layers kernel size of 3 has been used. Each encoder layer applies a sequence of transformations to the input tensor x, as defined below:

$$\begin{aligned} x &= \text{conv1}(x) \\ x &= \text{GN1}(x) \\ x &= \text{relu1}(x) \\ x &= \text{conv2}(x) \\ x &= \text{GN2}(x) \\ x &= \text{relu2}(x) \end{aligned} \tag{6.1}$$

where the details will be explained in the sequel.

M1 Bottleneck: At the core of the architecture, an LSTM layer with a hidden size of 64 serves as the bottleneck. This layer is crucial for capturing and integrating temporal dynamics of the signals, which is particularly vital for scenarios where interference exhibits time-dependent behaviors.

M1 Decoder: The decoder reverses the process of the encoder with two layers that progressively reconstruct the target signal from the obtained feature representations. The decoder layers include operations to combine the features from the encoder using skip connections, where skip denotes the corresponding encoder feature map, and are structured as follows:

$$\begin{aligned} x &= \text{convT}(x) \\ x &= \text{GN}(x) \\ x &= \text{relu1}(x) \\ x &= \text{skip\_handler}(x, \text{skip}) \\ x &= \text{conv}(x) \\ x &= \text{GN}(x) \\ x &= \text{relu2}(x) \end{aligned} \tag{6.2}$$

The encoder and decoder blocks, which form the fundamental building components of our models, are illustrated in fig. 6.1.

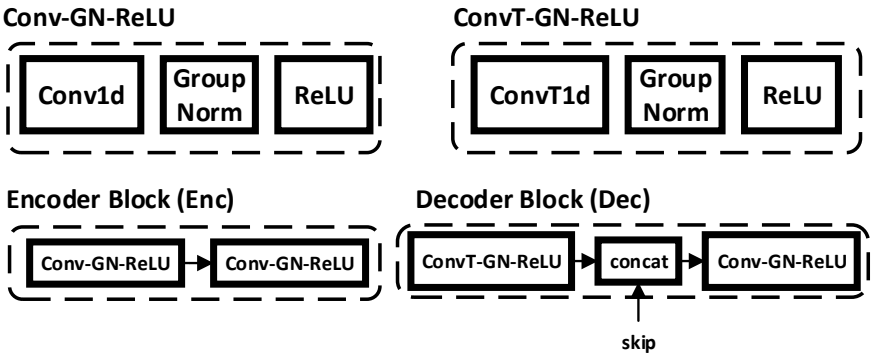


Figure 6.1: The architecture of the encoder and decoder blocks used in the models. The encoder block (Enc) consists of Conv1d, Group Normalization (Group Norm), and ReLU activation. The decoder block (Dec) mirrors the encoder but uses ConvT1d instead of Conv1d and includes concatenation (concat) for skip connections.

- **Conv1d**: Applies a 1-dimensional convolution over an input signal composed of several input planes. This operation effectively captures local dependencies within the input data.
- **GroupNorm**: Applies Group Normalization over a mini-batch of inputs. This normalization method divides the channels into groups (here we use groups of 4) and computes within each group the mean and variance for normalization, which stabilizes the learning process and improves the training dynamics.
- **LeakyReLU**: Applies the Leaky Rectified Linear Unit function element-wise. Unlike the standard ReLU, LeakyReLU allows a small, non-zero gradient when the unit is not active, which helps prevent neurons from dying during training.
- **ConvTranspose1d**: Applies a 1-dimensional transposed convolution operator over an input image. Commonly referred to as deconvolution, it is instrumental in upsampling the feature maps to higher resolutions.
- **Skip Handler**: Manages the skip connections between the encoder and decoder blocks. Depending on the configuration (concatenation and addition are most common operations), we use concatenation in our models, which is crucial for integrating learned features at various levels and recovering spatial information lost during downsampling.

These operations collectively contribute to the robust feature extraction and reconstruction capabilities of the U-Net CNN Model, ensuring detailed and accurate signal processing critical for effective interference cancellation.

M2 Architecture: The M2 model follows a similar architectural pattern to M1 but with key variations aimed at exploring different aspects of feature processing for CCI cancellation. Specifically, M2 omits the LSTM bottleneck and adjusts its layering to deepen the feature extraction capability. Both the encoder and decoder in M2 utilize the same double convolution structure as outlined for M1, but with modifications to the filter configuration and stride. The encoder features an extended sequence of layers with filters configured as [64, 128, 128, 128, 128, 128, 128, 128, 128] and corresponding strides of [1, 2, 2, 2, 2, 2, 2, 2, 2]. This configuration allows for deeper feature extraction while maintaining the model's efficiency through controlled spatial reduction in each strided convolution layer.

In both M1 and M2 models, the first convolution in each double convolution block of the encoder and decoder is responsible for adjusting the number of channels and applying strided convolution to reduce spatial or temporal dimensions. The second convolution focuses on enhancing feature representation without changing the dimensions. By excluding the LSTM layer, M2 emphasizes a purely convolutional approach. Fig. 6.2 presents the overall architecture of M1 and M2 models, showcasing the arrangement and integration of the encoder and decoder blocks with additional layers and skip connections.

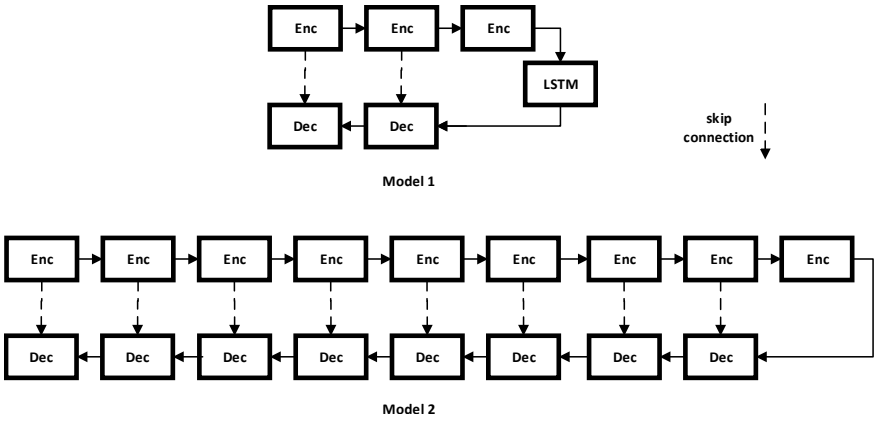


Figure 6.2: Model 1 (M1) utilizes a combination of encoder and decoder blocks with an LSTM layer and skip connections. Model 2 (M2) features a more extensive series of encoder and decoder blocks, incorporating multiple layers of skip connections for enhanced depth and complexity.

6.2.1.2 Depthwise Convolution

The depthwise convolution technique is integrated into the proposed models to optimize computational efficiency while maintaining the effectiveness of the feature extraction process. A depthwise convolution performs spatial convolution

independently over each input channel, reducing the computational load compared to standard convolutions [18]. This section details the structure of depthwise convolution blocks in both encoder and decoder segments of the network.

Encoder with Depthwise and Point-wise Convolutions: The encoder utilizes depthwise convolution followed by point-wise convolution to efficiently process the input signal. This arrangement allows for a significant reduction in computational complexity by separating the filtering and combining aspects of the convolution process. For input tensor x:

$$\begin{aligned} x &= \text{depthwise\_conv}(x) \\ x &= \text{pointwise\_conv}(x) \\ x &= \text{GN}(x) \\ x &= \text{relu1}(x) \\ x &= \text{depthwise\_conv2}(x) \\ x &= \text{pointwise\_conv2}(x) \\ x &= \text{GN}(x) \\ x &= \text{relu2}(x) \end{aligned} \tag{6.3}$$

The first depthwise convolution applies a spatial filter to each input channel independently, followed by a point-wise convolution that combines these filtered channels into the desired number of output channels. This two-step process efficiently manages feature extraction without excessively increasing the parameter count.

Decoder with Depthwise and Point-wise Convolutions: Similarly, the decoder employs depthwise and point-wise convolutions to reconstruct the signal from encoded features. It includes an additional step to handle the concatenated skip connections from the encoder, ensuring that spatial information is effectively reintegrated. For input tensor x:

$$\begin{aligned} x &= \text{depthwise\_convT}(x) \\ x &= \text{pointwise\_convT}(x) \\ x &= \text{GN}(x) \\ x &= \text{relu1}(x) \\ x &= \text{skip\_handler}(x, \text{skip}) \\ x &= \text{depthwise\_conv}(x) \\ x &= \text{pointwise\_conv}(x) \\ x &= \text{GN}(x) \\ x &= \text{relu2}(x) \end{aligned} \tag{6.4}$$

In the decoder, the depthwise convolutions first upsample the feature maps, followed by point-wise convolutions that adjust the channel dimensions to match

the subsequent layers. This method effectively preserves detailed features while efficiently managing computational resources.

6.2.1.3 Baseline Models from Prior Scientific Work

To evaluate the effectiveness of the proposed U-Net architectures, comparisons are made against two well-established baseline models in the field of signal separation: WaveNet and ConvTasNet. These models were chosen due to their proven capabilities and distinct approaches to handling similar tasks in both audio and RF signal processing domains.

WaveNet: Originally developed for audio synthesis and later adapted for source separation, WaveNet is known for its utilization of dilated convolutions, which allow it to capture a wide range of temporal contexts efficiently. In our study, the WaveNet baseline is configured with 30 residual layers, using 128 residual channels, and a dilation cycle length of 10. This setup is designed to optimize the model's ability to separate overlapping signals by expanding its receptive field without significant increases in computational complexity. This model also forms the main baseline performance to beat in ICASSP 2024 grand challenge [14] for known interference types.

ConvTasNet: ConvTasNet employs a convolutional approach to perform source separation through a masking technique, distinguishing it from other spectral-domain separation methods. For our purposes, ConvTasNet is configured with parameters as follows: $N = 512$, $L = 16$, $B = 128$, $H = 512$, $P = 3$, $X = 8$, $R = 3$. These parameters define the network's architecture, including the number of filters (N), the length of the filters in samples (L), bottleneck width (B), number of channels in convolutional blocks (H), kernel size of the convolutional blocks (P), number of convolutional blocks within each repeat (X), and the number of repeats (R). Detailed explanations of these parameters can be found in [19] as well as in the Conv-TasNet implementation on GitHub[1].

6.2.2 Quantization Techniques

Quantization-Aware Training (QAT) is employed to optimize the proposed models for efficient deployment, particularly focusing on convolutional layers. QAT simulates the quantization effects during training, incorporating fake quantization nodes within the model architecture. This allows the training to adjust for the quantization-induced errors and refine the model parameters accordingly.

Implementation of QAT: In our methodology, quantization is applied only to the convolutional layers using an 8-bit representation. This approach is selected due to its balance between performance and computational efficiency. The QAT process begins with the pre-trained unquantized model, which undergoes fine-tuning

[1] https://github.com/JusperLee/Conv-TasNet

with quantization emulations activated. This strategy leverages the pre-existing model weights and optimizes them under quantization constraints, which avoids the need for training from scratch and ensures the model's performance remains robust. The primary benefits of applying QAT include reduced model size and computational demands, making the models suitable for deployment on devices with limited computational resources (8 bit processors). Moreover, by incorporating quantization during the training phase, the models are better prepared for the reduced precision environment as compared to post-training quantization, which enhances their operational efficiency.

6.2.3 Dataset and Training

6.2.3.1 Dataset Description

The datasets utilized in this study are sourced from the ICASSP 2024 grand challenge [14], which offers a diverse set of interference types tailored for research in CCI cancellation. The dataset comprises four distinct interference categories:

- **EMISignal1:** Electromagnetic interference from man-made sources.
- **CommSignal2** and **CommSignal3:** Digital communication signals from commercial wireless devices.
- **CommSignal5G1:** A 5G-compliant waveform.

The variety of these datasets provides a robust testing ground for the study, as it covers a broad spectrum of potential real-world interference scenarios. Notably, the provided labels indicating interference type are not utilized in our models to adhere to the study's assumption of unknown interference sources.

6.2.3.2 Data Preparation

Data preparation involves converting complex-valued 1D signals into two real-valued channels, representing the in-phase (I) and quadrature (Q) components. Additionally, two forms of data augmentation are employed to enhance the robustness of the models under varying signal conditions:

1. **Random SINR adjustment:** The SINR is varied randomly between -30 dB and 0 dB by adjusting the gain of the interference signal.
2. **Random Phase Shift:** A random phase between $-\pi$ and π is added to the interference signal's phase before it is combined with the Signal of Interest (SOI).

The dataset distribution is detailed in terms of the number of superframes and the number of samples in each superframe for each type of interference, formatted as (number of superframes: number of samples):

- **CommSignal3:** 139: 260,000
- **CommSignal2:** 100: 43,560
- **CommSignal5G1:** 149: 230,000
- **EMISignal1:** 530: 230,000

A superframe is defined as the longest measurement window available in the dataset, from which smaller windows of length L are extracted for training and validation. Additionally, 50 superframes are explicitly reserved and separated from the training dataset for the sole purpose of final testing, ensuring an unbiased evaluation of the model performance. For training purposes, the training dataset is randomly divided into an 80% training set and a 20% validation set, based on the number of superframes. Each batch for training or validation is dynamically generated by randomly selecting a superframe from the dataset, choosing a random starting index within the superframe, and extracting a segment of length L.

To evaluate robustness, the model's performance was averaged over a range of SNR values between -30 dB and 0 dB, using steps of 3 dB, consistent with the ICASSP RF signal separation challenge. This approach ensures that the reported results reflect the models' ability to handle diverse signal conditions and varying levels of interference. Furthermore, the dataset comprises multiple interference types—5G-compliant waveforms, electromagnetic interference, and commercial wireless signals—without prior knowledge of interference characteristics, ensuring the models are robust to both complex and unknown scenarios.

6.2.3.3 Training Process

The training of the models is conducted using a batch size of 2, with an initial learning rate of 0.002 and an Adam optimizer. The learning rate is adjusted over the course of training using a cosine annealing scheduler, which helps in fine-tuning the model parameters towards the latter stages of training. The models are trained for 100,000 optimization steps with segments of data of length $L = 512$ extracted from the superframes as described. Training is performed on NVIDIA A40 GPUs, and the complete training process takes approximately 23 hours for our proposed models. This setup ensures that the models are thoroughly optimized across the varied scenarios presented in the training dataset.

6.2.3.4 Evaluation Metrics

The performance of the models is evaluated using the Mean Squared Error (MSE) score, a standard metric for quantifying the accuracy of models in regression tasks, which in this context measures the deviation of the model's outputs from the true signal values as detailed in subsec. 4.2.2.

To ensure stable and continuous optimization during training, we employed a smoothed version of the Mean MSE loss in dB. This custom loss function uses a sigmoid-shaped transition to avoid sharp gradients and promote better generalization. The smoothing function is defined as:

$$\text{Loss}(mse_{\text{dB}}) = L + \frac{U - L}{1 + \exp(-\alpha \cdot (mse_{\text{dB}} - M))}, \tag{6.5}$$

where:

- L: Lower bound of the MSE range (-50 dB).
- U: Upper bound of the MSE range (50 dB).
- α: Smoothness parameter controlling the sharpness of the transition (0.1).
- M: Midpoint of the range, defined as $M = \frac{L+U}{2}$.

The output of this function is clamped between the lower and upper bounds to ensure stability. By smoothing the loss in this manner, the model avoids extreme gradients that could destabilize training while maintaining focus on achieving uniformly good reconstructions. This contributes to balanced optimization and enhances the overall robustness of the network.

6.3 Results

In this section, we present the results of our evaluations to demonstrate the performance and computational efficiency of the proposed architectures compared to the baseline models. Our analysis includes various aspects such as MSE score, computational complexity, inference speed, memory footprint, and parallelizability on GPUs.

We begin by examining the relationship between the MSE score and the number of MACs. This analysis allows us to evaluate the performance and computational resources required for each of the two proposed architectures as well as the two baseline models. Following this, we present the MSE score versus symbol rate of the model, which measures the inference speed and indicates how fast each architecture operates on both CPU and GPU. This is crucial for understanding the real-time applicability of the models. Next, we provide a summary of the models, including their MSE scores, MACs, number of parameters, and size or memory footprint. We also discuss the impact of applying depthwise convolution and quantization techniques on these metrics. Furthermore, we analyze the symbol rate of different models on a GPU for various batch sizes during inference. This analysis showcases the parallelizability of the architectures on GPUs, highlighting their efficiency in handling large-scale data processing. Lastly, we explore the

effects of applying pruning to the models to investigate if further compression and efficiency gains can be achieved through this method.

6.3.1 Model Performance

6.3.1.1 Performance and Complexity

To evaluate the performance and computational efficiency of our models, we present a scatter plot of the MSE score versus the number of MACs in fig. 6.3. This plot includes our two proposed architectures (M1 and M2) and two baseline models (ConvTasNet and WaveNet). The size of the circles in the plot is proportional to the number of parameters of each model, providing a visual representation of model complexity (computational as well as memory).

From fig. 6.3, we observe the following. M1 and M2 achieve higher MSE scores compared to the baseline models, indicating superior performance in signal separation. Additionally, they require significantly fewer MACs, demonstrating lower computational complexity. The sizes of the circles for M1 and M2 indicate that both have fewer parameters than the baseline models, with M1 having slightly fewer parameters and performing slightly better in terms of MSE compared to M2.

ConvTasNet has a lower MSE score than M1 and M2, implying less effective signal separation. It also requires more MACs. The larger circle size indicates a higher number of parameters compared to M1 and M2. WaveNet has the lowest MSE score among all models, indicating the least effective performance in terms of signal separation. It demands the highest number of MACs and parameters, as shown by the large circle size. These findings underscore the trade-off between model performance and computational complexity. Although WaveNet requires substantial computational resources and has a high model complexity, it does not provide better signal separation compared to our proposed architectures. In contrast, M1 and M2 strike a balance by achieving superior signal separation with significantly lower computational demands, making them more suitable for deployment in resource-constrained environments.

Next, we evaluate the inference rate of these models to further understand their efficiency and practical applicability.

6.3.1.2 Inference Rate

Inference speed is a critical component of the end-to-end latency in communication systems, which also includes signal acquisition, pre-processing, and post-processing delays. This study focuses on minimizing the inference time by optimizing neural network architectures. While other latency components depend on hardware and system-level factors, reducing inference time directly addresses a key bottleneck, contributing to lower overall communication latency in real-time systems.

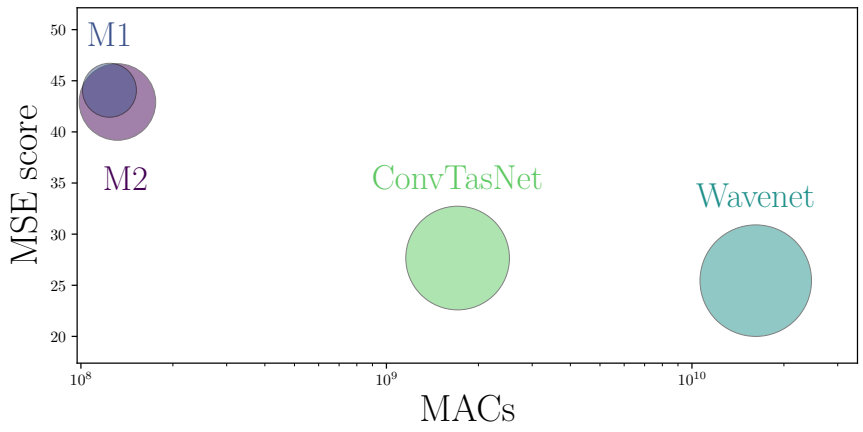


Figure 6.3: Scatter plot of MSE score versus MACs for different models. The size of the circles is proportional to the number of parameters of each model.

The inference rate of the models is evaluated on both CPU and GPU, using the following hardware configurations:

- **CPU:** Intel(R) Xeon(R) Silver 4310 CPU @ 2.10GHz with 48 cores.
- **GPU:** NVIDIA A40 with 46,068 MiB memory, CUDA Version 12.2.

The symbol rate is measured as the number of communication symbols that can be decoded when performing interference cancellation in one second. It is an important practical metric, as a low symbol rate can become a bottleneck for the end-to-end communication system, affecting overall system performance.

Before analyzing the symbol rate results, it is important to clarify a key point. Unlike MACs, inference time is influenced not only by the model architecture and its theoretical computational complexity but also by the platform, which includes both the hardware (CPU or GPU) and the software and libraries that determine how layers (e.g., convolution) are implemented. This means that a model with fewer operations (in terms of MACs) can still be slower depending on its implementation. In this work, we use PyTorch version 2.2.1+cu121 for both training and inference time measurements. The inference time measurements presented in this part of the study use a batch size of 1.

From fig. 6.4, we observe the following: On the CPU, M1 and M2 both achieve higher symbol rates compared to the baseline models, with M1 slightly outperforming M2. This indicates that our proposed architectures are more efficient in terms of inference speed on the CPU. As expected, having fewer MACs, both M1 and M2 demonstrate superior performance due to their efficient implementation.

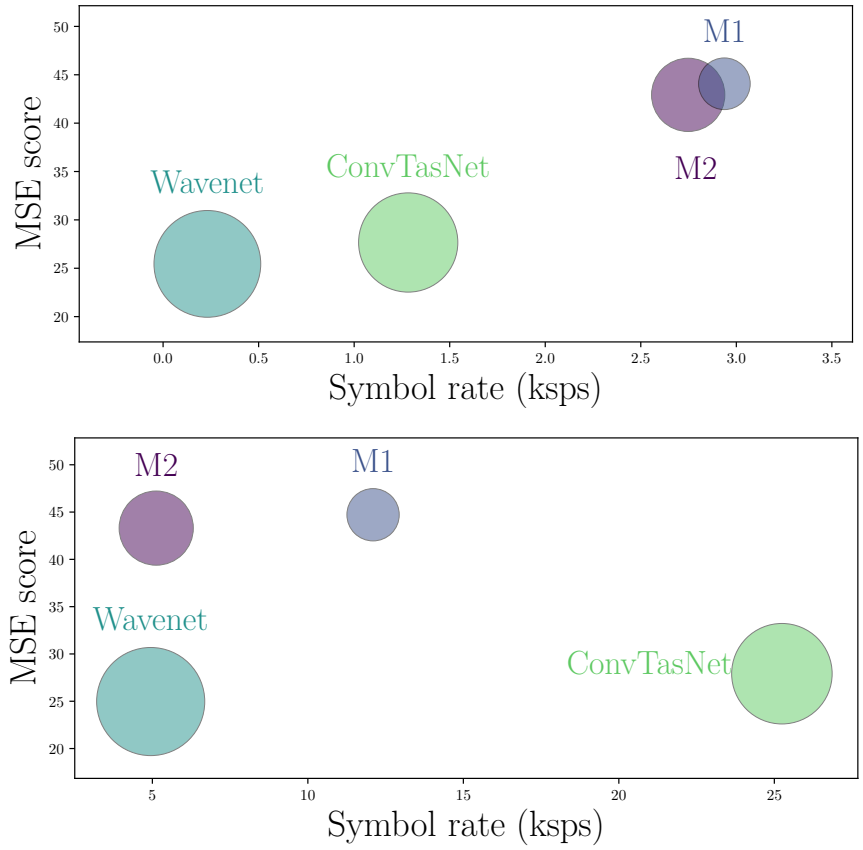


Figure 6.5: Performance comparison of different models on a GPU in terms of MSE score and symbol rate (ksps) with an inference batch size of one. The size of each bubble represents the model's number of parameters. On the GPU, ConvTasNet achieves a high symbol rate with a competitive MSE score, while M1 and M2 demonstrate high symbol rates and strong MSE scores. Wavenet shows moderate performance in both metrics.

However, on the GPU, ConvTasNet outperforms all other models in terms of symbol rate for inference batch size of one. This suggests that ConvTasNet benefits significantly from the parallel processing capabilities of the GPU (we will show it is not the case for larger batch sizes in sec. 6.3.3). Additionally, M1 and M2 show a notable divergence in performance; M1 outperforms M2 in both symbol rate and MSE score. This highlights the superior scalability and efficiency of M1 when leveraging GPU resources compared to M2. These observations underscore the importance of considering both computational complexity and platform-specific implementation when evaluating model performance. While our proposed models (M1 and M2) are highly efficient on the CPU, ConvTasNet demonstrates exceptional

performance on the GPU, making it a viable option for high-throughput applications when a lower performance can be tolerated.

6.3.2 Comparative Analysis

As shown in table 6.1, we provide a summary of the models' MSE scores, MACs, number of parameters, and memory footprint (SizeMb) with and without applying depthwise convolution (Dw) and quantization (Q). The observed performance

Table 6.1: Summary of Model Metrics

Model	MSE score	MACs	Parameters	SizeMb
M1	**44.07**	123,994,112	927,874	3.78
M1(Dw)	43.75	51,252,224	**514,570**	2.13
M1(Q)	33.10	123,994,112	927,874	**1.90**
M2	42.92	131,694,592	1,866,370	7.56
M2(Dw)	43.19	**51,231,232**	646,922	2.70
M2(Q)	31.25	131,694,592	1,866,370	2.02
WaveNet	25.45	16,175,333,376	3,964,674	15.92
ConvTasNet	27.66	1,710,026,752	3,425,458	13.83

changes when applying depthwise (Dw) convolutions to U-Net architectures can be explained by the distinct roles of their layers. In the pure convolutional U-Net (M2), the reduction in parameters from Dw convolutions mitigates overfitting and promotes more efficient feature extraction, leading to a slight performance improvement from an MSE score of 42.92 to 43.19. Conversely, in the U-Net with LSTM (M1), the LSTM layer depends on rich, high-dimensional feature maps to capture temporal dependencies effectively. Depthwise convolutions reduce the richness and informativeness of these feature maps, impairing the LSTM's ability to model temporal patterns, resulting in a slight performance degradation from an MSE score of 44.07 to 43.75.

Quantization, which reduces precision to 8 bits, has a more severe impact on performance compared to using depthwise convolutions. The MSE score drops significantly to 33.10 for M1 and to 31.25 for M2. However, quantization does not reduce the number of operations, and hence the MACs remain unchanged.

The SizeMb column in table 6.1 indicates the memory footprint required to store the model's weights. Depthwise convolutions reduce the MACs to less than half in both M1 and M2, significantly decreasing the memory footprint to 2.13 MB for M1(Dw) and to 2.70 MB for M2(Dw). Quantization further reduces the model size to 1.90 MB for M1(Q) and to 2.02 MB for M2(Q).

These results highlight the trade-offs between model size, computational complexity, and performance. While our proposed architectures (M1 and M2) maintain a balance of high performance and low computational demands, applying depthwise convolutions and quantization can further optimize their memory footprint and computational efficiency. This makes them more suitable for deployment in resource-constrained environments while still achieving effective signal separation.

6.3.3 Parallelizability on GPU

Next, we analyze the symbol rate of different models on a GPU for various batch sizes during inference to showcase the parallelizability of the architectures. To evaluate the parallelizability of the models, we analyzed their performance on a GPU (NVIDIA A40 with 46,068 MiB memory, CUDA Version 12.2) across different batch sizes. The symbol rate (Ksps) as a function of batch size is depicted in fig. 6.6. The purpose of this analysis is to determine how effectively each architecture can leverage the GPU's capability to process multiple batches of input simultaneously. It is important to note that we do not report quantization results for GPU since 8-bit precision computation is not typically supported on GPUs. Consequently, the (Q) models are omitted in this analysis.

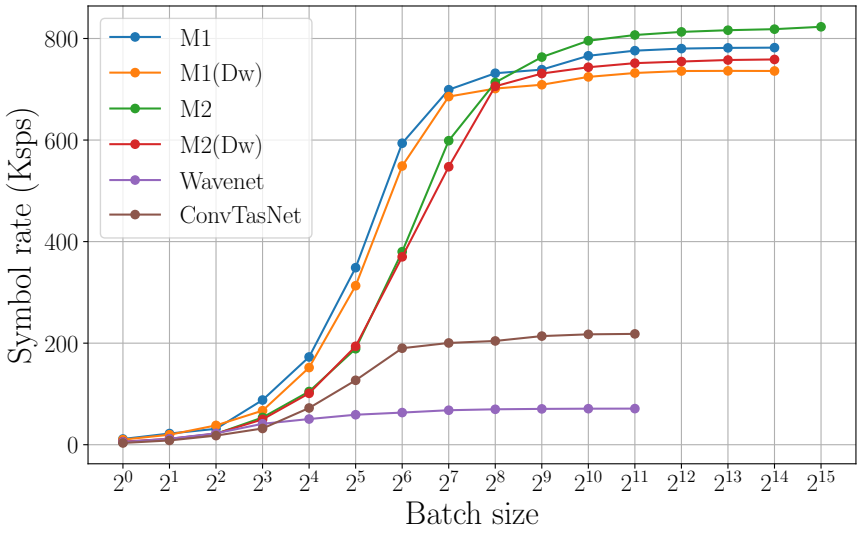


Figure 6.6: Inference throughput as a function of batch size, illustrating scalability and efficiency gains during batch processing. Each curve extends until the model encounters an Out-Of-Memory (OOM) error, resulting in different stopping points across models.

From fig. 6.6, several key observations can be made:

- Initially, M1 and M1(Dw) achieve a higher symbol rate compared to M2. However, as the batch size increases beyond 2^{10}, the purely convolutional model M2 reaches the highest symbol rate at large batch sizes. This indicates that while models with LSTM layers (M1 and M1(Dw)) perform well at smaller batch sizes, fully convolutional models (M2 and M2(Dw)) scale better with increasing batch sizes.

- WaveNet and ConvTasNet encounter Out-of-Memory (OOM) errors at smaller batch sizes compared to M1, M1(Dw), M2, and M2(Dw), whereas M2 can handle the largest batch sizes before encountering OOM errors. This demonstrates the superior scalability of M2 in terms of batch processing on the GPU.

These results provide valuable insights for applications demanding high throughput or low latency. In such scenarios, the ability to process larger batch sizes effectively can significantly impact overall performance. As expected, models incorporating LSTM layers (M1 and M1(Dw)) do not scale as well as fully convolutional models (M2 and M2(Dw)). Therefore, for applications where large batch processing and scalability are crucial, purely convolutional models like M2 are more advantageous. In the following section, we explore the potential for further model compression through pruning and its impact on performance.

6.3.4 Impact of Pruning

Pruning techniques, such as structured and unstructured pruning, are widely used to reduce the size and computational complexity of machine learning models. Structured pruning removes entire filters, channels, or layers, resulting in direct computational benefits and compatibility with hardware acceleration. Conversely, unstructured pruning removes individual weights, leading to sparse weight matrices that reduce memory footprints but typically offer minimal computational gains on most hardware platforms.

In this study, we focused on unstructured pruning, which removes weights from convolutional and transposed convolutional layers based on their norm (e.g., pruning the weights with the lowest l_1-norm). While this method allows finer granularity in parameter reduction, our results indicate significant performance degradation at higher pruning ratios with limited computational benefits. This highlights the sensitivity of our models to parameter reduction due to extensive hyperparameter tuning during training, which was designed to optimize model efficiency under fully-parameterized conditions. Future work will explore structured pruning strategies to achieve a better trade-off between performance and computational efficiency.

To evaluate the potential for further model compression, we applied unstructured pruning at various ratios and measured the resulting MSE scores. Pruning is a

technique used to reduce the number of parameters in a model, thereby decreasing its size. However, it can also degrade model performance if the pruned parameters are crucial for maintaining accuracy.

In this work, we apply unstructured pruning which removes individual weights from the convolutional and transposed convolutional layers based on their norm (pruning by removing filters with the lowest l_1-norm), without considering their spatial positions. This type of pruning can lead to sparse weight matrices, but does not necessarily translate to faster inference since most hardware and software frameworks are optimized for dense computations.

Table 6.2: MSE Scores for Different Models and Pruning Ratios

Model	Pruning Ratio				
	0.01	0.05	0.10	0.25	0.50
M1	31.67	30.32	25.69	15.94	6.51
M1(Dw)	30.59	26.67	20.49	10.21	2.14
M2	29.62	28.85	24.34	16.19	7.42
M2(Dw)	28.05	25.50	20.30	10.79	2.43
Wavenet	12.51	12.54	12.34	10.50	6.14
ConvTasNet	14.12	13.67	12.17	6.88	1.96

Table 6.2 presents the MSE scores for different models and pruning ratios. From these results, we observe a general trend of performance degradation as the pruning ratio increases. This is expected, as higher pruning ratios remove more parameters, which can negatively impact the model's ability to accurately separate signals.

For M1, the MSE score decreases from 31.67 to 6.51 as the pruning ratio increases from 0.01 to 0.50. Similarly, M1(Dw) shows a significant drop in performance, with the MSE score decreasing from 30.59 to 2.14. This suggests that M1 and M1(Dw) are highly sensitive to pruning, likely due to their reliance on rich feature representations to capture temporal dependencies.

M2 and M2(Dw) exhibit a similar sensitivity to pruning. The MSE score for M2 drops from 29.62 to 7.42, while M2(Dw) decreases from 28.05 to 2.43. The pure convolutional architecture (M2) and its depthwise variant also rely on a large number of parameters to perform effective feature extraction, which explains their performance degradation with increased pruning.

Additionally, the reference models ConvTasNet and Wavenet have been included to offer a comparative perspective. ConvTasNet starts with an MSE of 14.12 at the lowest pruning ratio of 0.01, dropping to 1.96 at 0.50, showing its robustness compared to other models. Wavenet begins with an MSE of 12.51 at 0.01 and decreases to 6.14 at 0.50, indicating a relatively stable performance across varying pruning levels.

The pronounced decline in MSE scores across all models underscores the

substantial impact of pruning on performance. This is particularly notable when compared with the optimal scores previously reported, highlighting a significant degradation. The sensitivity observed can be attributed to the extensive hyperparameter tuning tailored to maximize model efficiency under fully-parameterized conditions, inadvertently rendering the models less tolerant to the reduction of parameters.

In practical applications, this sensitivity implies that while some degree of model compression is feasible, pruning in non-overparameterized models can lead to considerable performance losses. Consequently, careful consideration must be given to the trade-off between model size reduction and performance degradation when employing pruning techniques for model compression.

This concludes our evaluation of model performance, inference rate, parallelizability, and the impact of pruning. These comprehensive analyses provide valuable insights into the strengths and limitations of the proposed architectures and their suitability for real-world deployment.

6.4 Discussion

6.4.1 Analysis of Results

The results presented in this study provide an evaluation of the proposed architectures (M1 and M2) compared to the baseline models (ConvTasNet and WaveNet). Our analysis indicates that M1 and M2 achieve higher MSE scores with significantly fewer MACs, demonstrating their efficiency in terms of computational complexity. Specifically, M1 and M2 showed strong performance in signal separation while maintaining a lower computational footprint, making them suitable for deployment in resource-constrained environments.

When evaluating the inference rate on the GPU, we observed that M1 initially achieves higher symbol rates at smaller batch sizes. However, as batch size increases, M2 outperforms M1, indicating better scalability and parallelizability. ConvTasNet, despite its higher computational demands, demonstrates exceptional performance in a single batch setup on GPU. These results highlight the importance of considering both computational complexity and implementation efficiency when selecting models for real-time applications.

The application of depthwise convolutions and quantization further reduced the memory footprint and computational demands of M1 and M2. However, this came at the cost of some performance degradation. Depthwise convolutions, in particular, had a varied impact, slightly improving the performance of M2 while degrading M1. Quantization led to a more significant drop in performance, underscoring the trade-offs involved in model optimization techniques.

The pruning results revealed that our models are sensitive to parameter reduction,

likely due to extensive hyperparameter tuning during their design. The significant performance degradation observed with higher pruning ratios suggests that pruning may not be the most effective strategy for optimizing these models for faster inference.

6.4.2 Deployment Considerations

Given the focus on edge devices, we have expanded our discussion to include deployment-related considerations such as power consumption, latency, and hardware compatibility. Power consumption is linked to the number of MACs, as established in studies like [20], and our proposed models significantly reduce MACs through the use of depthwise separable convolutions and quantization, achieving up to a 60% reduction in computational demands. Latency has been evaluated in terms of inference speed, with the fully convolutional model (M2) demonstrating superior scalability and throughput, achieving up to $800{,}000$ symbols per second on GPUs. Additionally, hardware compatibility is addressed through quantization-aware training (QAT), which reduces model size (e.g., M1(Q): 1.90 MB) and ensures efficient deployment on low-precision devices without significant performance degradation. These considerations underscore the practical viability of our models for real-time applications on resource-constrained platforms.

6.4.3 Implications and Future Work

The findings from this study have several practical implications. The superior performance and lower computational complexity of M1 and M2 make them attractive for applications where computational resources are limited. Their ability to maintain high MSE scores with fewer MACs highlights their potential for efficient real-time signal processing in communication systems.

The varying scalability of the models on the GPU suggests that model selection should be tailored to the specific requirements of the deployment environment. For instance, applications demanding high throughput and low latency may benefit more from M2, given its superior performance at larger batch sizes.

Future work will focus on further optimizing these models for deployment. This includes exploring structured pruning techniques that may offer a better trade-off between performance and computational efficiency. Additionally, investigating more advanced quantization methods that minimize performance loss while reducing model size will be a key area of research. Enhancing the implementation of sparse operations to leverage hardware capabilities more effectively can also provide significant performance improvements.

Moreover, extending the evaluation to include other relevant metrics, such as Bit Error Rate (BER) analysis, will provide a more comprehensive understanding of the models' performance in practical communication scenarios. Finally, adapting

these models to different hardware platforms and exploring their potential in other signal processing applications will be crucial for broadening their applicability.

6.4.4 Future Work

While this study focuses on QPSK modulation, the proposed methodologies are not inherently restricted to this modulation scheme. The network architecture and optimization techniques, including depthwise separable convolutions and quantization, can be adapted to other modulation formats, such as QAM. Extending the models to support additional modulation schemes would require re-training with appropriate datasets that reflect the characteristics of the target modulation types. Exploring these scenarios represents a valuable direction for future research, particularly in heterogeneous communication environments where multiple modulation formats coexist. Furthermore, embedding modulation-specific information into the network architecture could enhance the model's adaptability to diverse signal environments.

While this study evaluates quantization, depthwise separable convolutions, and pruning individually to establish their viability, we have not explored all combinations of these techniques. Future research could investigate integrating the most promising methods, such as quantization and depthwise separable convolutions, to determine if their benefits can be combined effectively with acceptable performance trade-offs.

Another interesting direction for future work involves studying the generalization capabilities of the models to out-of-distribution interference types. Preliminary experiments during the challenge showed minor performance degradation for most unseen interference types, except for CommSignal2, which posed significant challenges. A more comprehensive study could investigate the relationship between the characteristics of interference signals and the models' ability to generalize, potentially providing insights into the inherent difficulty of separating certain interference types.

6.5 Conclusion

In this paper, we have presented an in-depth evaluation of two proposed U-Net architectures (M1 and M2) for signal separation, comparing them with baseline models (ConvTasNet and WaveNet). Our results demonstrate that M1 and M2 achieve superior performance in terms of MSE scores while maintaining significantly lower computational complexity, making them suitable for deployment in resource-constrained environments.

Specifically, M1, which incorporates depthwise separable convolutions and an LSTM, reduces MACs by 58.66% with only a 0.72% degradation in MSE score. M2, a fully convolutional model, when implemented by depthwise separable

convolutions achieves a 0.63% improvement in MSE score while using 61.10% fewer MACs compared to original M2 models. These substantial reductions in computational complexity highlight the efficiency of the proposed models.

Our analysis of inference rates on both CPU and GPU indicates that M2 scales particularly well with larger batch sizes, achieving symbol rates of up to 800,000 symbols per second. This scalability makes M2 an excellent choice for applications demanding high throughput and low latency. The optimization techniques, including quantization and depthwise separable convolutions, effectively reduce memory footprint and computational demands, though some performance trade-offs are observed. Pruning, especially unstructured pruning, demonstrated the models' sensitivity to parameter reduction, suggesting the need for more sophisticated optimization strategies.

These findings provide valuable insights into selecting and optimizing models for real-time signal processing applications. Future research will explore structured pruning techniques, advanced quantization methods, and leveraging hardware capabilities for sparse operations. Additionally, extending the evaluation to include BER analysis and adapting the models to various hardware platforms will further enhance their practical applicability.

In conclusion, our proposed architectures achieve a promising balance between performance and efficiency, positioning them as strong candidates for advanced signal separation tasks in communication systems and beyond.

References

[1] A. Défossez, N. Usunier, L. Bottou, and F. Bach. *Music Source Separation in the Waveform Domain*. arXiv preprint arXiv:1911.13254, 2019.

[2] S. Asadzadeh, T. Yousefi Rezaii, S. Beheshti, A. Delpak, and S. Meshgini. *A Systematic Review of EEG Source Localization Techniques and Their Applications on Diagnosis of Brain Abnormalities*. Journal of Neuroscience Methods, 339:108740, 2020. doi:https://doi.org/10.1016/j.jneumeth.2020.108740.

[3] R. Szupiluk, T. Ząbkowski, and T. Soboń. *Analysis of Financial Time Series Morphology with Amuse Algorithm and its Extensions*. Acta Physica Polonica A, 129(5):1018–1022, 2016.

[4] A. Boulais, O. Berné, G. Faury, and Y. Deville. *Unmixing Methods Based on Nonnegativity and Weakly Mixed Pixels for Astronomical Hyperspectral Datasets*. Astronomy & Astrophysics, 647:A105, 2021.

[5] M. S. Ibrahim, P. A. Karakasis, and N. D. Sidiropoulos. *A Simple and Practical Underlay Scheme for Short-Range Secondary Communication*. IEEE Transactions on Wireless Communications, 21(11):9990–10004, 2022. doi:10.1109/TWC.2022.3181618.

[6] K. S. Bondada, H. Yang, X. Cheng, D. J. Jakubisin, N. Tripathi, G. Anderson, Y. Yang, and J. H. Reed. *Enhanced Non-Preemptive Support of URLLC Using Spread Spectrum Underlay Signalling*. In Proc. IEEE Mil. Commun. Conf. (MILCOM)., pages 298–303, 2023. doi:10.1109/MILCOM58377.2023.10356263.

[7] T.-H. Vu, T.-V. Nguyen, D. B. da Costa, and S. Kim. *Performance Analysis and Deep Learning Design of Underlay Cognitive NOMA-Based CDRT Networks with Imperfect SIC and Co-Channel Interference*. IEEE Transactions on Communications, 69(12):8159–8174, 2021. doi:10.1109/TCOMM.2021.3110209.

[8] K. J. Silva Lorraine and M. Ramarakula. *A Comprehensive Survey on GNSS Interferences and the Application of Neural Networks for Anti-Jamming*. IETE Journal of Research, 69(7):4286–4305, 2023.

[9] M. Naseri, J. Fontaine, I. Moerman, E. De Poorter, and A. Shahid. *A U-Net Architecture for Time-Frequency Interference Signal Separation of RF Waveforms*. In Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 2024.

[10] L. Henneke. *Improving Data-Driven RF Signal Separation with SOI-Matched Autoencoders*. In Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 2024.

[11] C. Yapar, F. Jaensch, J. C. Hauffen, F. Pezone, P. Jung, S. K. Dehkordi, and G. Caire. *DEMUCS for Data-Driven RF Signal Denoising*. In Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 2024.

[12] R. M. Buehrer, S. P. Nicoloso, and S. Gollamudi. *Linear Versus Non-Linear Interference Cancellation*. Journal of Communications and Networks, 1(2):118–133, 1999. doi:10.1109/JCN.1999.6596755.

[13] B. Clerckx and C. Oestges. *Chapter 12 - Multi-User MIMO*. In B. Clerckx and C. Oestges, editors, MIMO Wireless Networks (Second Edition), pages 419–523. Academic Press, Oxford, second edition edition, 2013. doi:https://doi.org/10.1016/B978-0-12-385055-3.00012-2.

[14] T. Jayashankar, B. Kurien, A. Lancho, G. C. Lee, Y. Polyanskiy, A. Weiss, and G. W. Wornell. *The Data-Driven Radio Frequency Signal Separation Challenge*. In Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 2024.

[15] M. Naseri, E. de Poorter, I. Moerman, H. V. Poor, and A. Shahid. *Blind Co-Channel Interference Cancellation Using Fast Fourier Convolutions*. In 2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring), pages 1–2. IEEE, 2024.

[16] Z. Meng, Z. Fan, Z. Zhao, and F. Su. *ENS-Unet: End-to-End Noise Suppression U-Net for Brain Tumor Segmentation*. In Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC), pages 5886–5889, 2018. doi:10.1109/EMBC.2018.8513676.

[17] Y.-J. Cha, A. Mostafavi, and S. S. Benipal. *DNoiseNet: Deep Learning-Based Feedback Active Noise Control in Various Noisy Environments*. Engineering Applications of Artificial Intelligence, 121:105971, 2023.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. *Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arXiv preprint arXiv:1704.04861, 2017.

[19] Y. Luo and N. Mesgarani. *Conv-Tasnet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation*. IEEE/ACM Trans. Audio, Speech, Lang. Process., 27(8):1256–1266, 2019.

[20] T.-J. Yang, Y.-H. Chen, J. Emer, and V. Sze. *A method to estimate the energy consumption of deep neural networks*. In 2017 51st Asilomar Conference on Signals, Systems, and Computers, pages 1916–1920, 2017. doi:10.1109/ACSSC.2017.8335698.

Part III

Learned Image Compression for Noisy Channel Transmission

7

Learned Image Compression for Noisy Channel Transmission

Efficient image transmission in wireless networks demands a careful balance between reliability, throughput, and latency—particularly in the face of varying channel conditions. Traditional compression techniques often struggle to adapt to dynamic scenarios, where severe channel impairments can degrade visual quality or increase transmission delays. In this chapter, we introduce an adaptive and progressive learned image compression (LIC) framework, evaluated on two leading deep models—a hyperprior design and a Vector Quantized Generative Adversarial Network (VQGAN). Our approach aims to deliver flexible, high-performance image transmission even under challenging wireless environments.

In wireless communications, efficient image transmission must balance reliability, throughput, and latency, especially under dynamic channel conditions. This paper presents an adaptive and progressive pipeline for learned image compression (LIC) architectures tailored to such environments. We investigate two state-of-the-art learning-based models: the hyperprior model and the Vector Quantized Generative Adversarial Network (VQGAN). The hyperprior model achieves superior compression performance through lossless compression in the bottleneck but is susceptible to bit errors, necessitating error correction or retransmission mechanisms. In contrast, the VQGAN decoder demonstrates robust image reconstruction capabilities even without channel coding, enhancing reliability in challenging transmission scenarios. We propose progressive versions of both models, enabling partial image transmission and decoding under imperfect channel conditions. This

progressive approach not only maintains image integrity under poor channel conditions but also significantly reduces latency by allowing immediate partial image availability. We evaluate our pipeline using the Kodak high-resolution image dataset under a Rayleigh fading wireless channel model simulating dynamic conditions. The results indicate that the progressive transmission framework enhances reliability and reduces latency while maintaining or improving throughput compared to non-progressive counterparts, particularly in low Signal-to-Noise Ratio (SNR) environments. Specifically, the progressive hyperprior model consistently outperforms others in latency metrics, especially in the 99.9th percentile waiting time—a measure indicating the maximum waiting time experienced by 99.9% of transmission instances—across low SNRs, and achieves higher throughput where adaptive WebP fails. Adaptive WebP, an extension of the WebP image format that dynamically adjusts compression parameters based on channel conditions, demonstrates superior image quality and throughput at higher SNRs when channel conditions are favorable. The proposed method is particularly advantageous for Internet of Things (IoT) applications and task-oriented communications, where initial low-quality images suffice for immediate processing and subsequent data refinement enhances performance as conditions permit.

At the core of this chapter is a discussion on how an adaptive and progressive approach to learned image compression can flexibly respond to fluctuating channel conditions. While the hyperprior-based method excels in compression efficiency, it requires added mechanisms (e.g., error correction) for bit-error-prone transmissions. On the other hand, the VQGAN model offers inherent robustness to transmission errors, albeit with potential trade-offs in exact reconstruction fidelity.

A key innovation involves partial image decoding, whereby initial low-fidelity representations are made available at the receiver under suboptimal channel conditions, reducing user-perceived latency. Detailed evaluations highlight how these progressive strategies maintain or improve throughput in challenging Signal-to-Noise Ratio (SNR) environments, outperforming non-progressive baselines. We further explore the integration of adaptive WebP as a comparative reference point, noting that it achieves higher image quality and throughput under favorable SNRs but lacks the robustness and flexibility offered by learned progressive models in more adverse conditions.

The methods presented here hold particular promise for Internet of Things (IoT) use cases and task-oriented communication paradigms, where partial or incremental image information can be of immediate utility. By enabling refinement as conditions permit, the proposed architectures harmonize the competing demands of reliability, throughput, and latency. Subsequent sections will detail the design considerations behind these models, their training procedures, and experimental insights gleaned from the Kodak dataset under realistic Rayleigh fading channels.

⋆ ⋆ ⋆

This Chapter is adapted from:

Mostafa Naseri, Pooya Ashtari, Mohamed Seif, Eli De Poorter, H Vincent Poor, Adnan Shahid
Deep Learning-Based Image Compression for Wireless Communications: Impacts on Reliability, Throughput, and Latency

Submitted to IEEE Journal on Selected Areas in Communications, Intelligent Communications for Real-Time Computer Vision (Comm4CV), 2025.

This chapter targets C1–C3 fully and C4 partially.

7.1 Introduction

Efficient image transmission has become increasingly critical in modern wireless communication systems, driven by the rapid increase of applications that demand high-quality visual data even under challenging channel conditions. Applications such as remote sensing, autonomous driving, augmented reality, and Internet of things (IoT) rely on the timely and reliable transmission of images to function effectively [1, 2]. These scenarios often operate in environments with limited bandwidth and high levels of interference, necessitating robust and efficient image compression techniques.

Traditional image compression methods, including JPEG2000 [3], WebP [4], and BPG [5], are widely utilized due to their established performance and compatibility. However, these conventional techniques encounter significant limitations at extremely low bitrates. At such low bitrates, traditional compressors are compelled to employ large quantization steps, resulting in substantial degradation of image quality. Furthermore, these methods are highly sensitive to bit errors; even minor errors can render the entire image undecodable rather than merely degrading its quality. This sensitivity necessitates the use of additional error correction or error detection and retransmission mechanisms to ensure image integrity, thereby increasing latency and reducing overall system efficiency [6].

Noisy wireless channels, particularly those affected by additive white Gaussian noise (AWGN), intensify the challenges associated with image transmission. High bit error rates and limited bandwidth can severely degrade image quality, making it difficult to achieve acceptable performance at very low bits per pixel (BPP). While channel coding techniques can mitigate some errors, relying solely on them may not be sufficient or efficient for ultra-low bitrate image transmission. In such scenarios, achieving both high compression efficiency and robustness to noise becomes paramount, as traditional methods struggle to meet these dual requirements [7].

Recent advancements in deep learning-based compression and reconstruction methods offer promising solutions to these challenges. By leveraging deep learning architectures, these techniques can achieve significant compression while maintaining or even enhancing perceptual image quality [2, 8, 9]. Learned image compression (LIC) models, such as hyperprior models [9, 10] and vector quantized generative adversarial networks (VQGAN) [11], have demonstrated superior performance compared to traditional methods. While hyperprior models focus on quantization and lossless compression in the bottleneck of their autoencoders, VQGAN was originally developed as an image tokenizer for image generation transformers. Nonetheless, its utilization of vector quantization in the latent space facilitates robust image transmission even in the absence of channel coding, which is advantageous for real-time image transmission as it reduces latency and computational overhead associated with encoding and decoding processes. This characteristic makes VQGAN a valuable component in image transmission systems, especially under dynamic and poor channel conditions where traditional compression models may falter.

The integration of recent advancements in neural joint source-channel coding (JSCC) further underscores the progress and ongoing challenges in efficient image transmission over wireless channels. Notably, Yang *et al.* [12] introduce a neural JSCC backbone based on the Swin Transformer architecture, which demonstrates remarkable adaptability to diverse channel conditions and transmission requirements. Their approach incorporates a code mask module that prioritizes channel importance, enabling adaptive transmission through a single, unified model. This is achieved by integrating the target rate into multiple layers of the image compressor. In contrast, our methodology leverages a hyperprior model that organizes feature maps in a sorted manner, facilitating a different form of adaptability. Additionally, while Yang *et al.* focus primarily on the inference time of the encoder and decoder, our study focuses on the waiting time induced by dynamic channel conditions, providing a more comprehensive evaluation of system latency.

Furthermore, Wu *et al.* [13] present JSCCformer-f, a wireless image transmission paradigm that capitalizes on feedback from the receiver to enhance transmission efficacy. The unified encoder in JSCCformer-f effectively utilizes semantic information from the source image, channel state information, and the decoder's current belief about the source image derived from the feedback signal to generate coded symbols dynamically at each transmission block. However, similar to Yang *et al.*, their work primarily addresses the inference time of the model without delving into the waiting time associated with image transmission. Moreover, JSCCformer-f necessitates continuous feedback from the decoder, which introduces additional communication overhead and complexity. Unlike their approach, our research focuses on progressive transmission and provides an in-depth analysis of transmission waiting time, an important factor in delay sensitive applications, eliminating

the dependency on receiver feedback and thereby streamlining the transmission process.

These recent studies by Yang *et al.* [12] and Wu *et al.* [13] highlight significant strides in the development of flexible and efficient neural JSCC frameworks. However, they also reveal critical areas that remain unexplored, particularly concerning the comprehensive assessment of transmission latency and the optimization of progressive transmission strategies without relying on receiver feedback. Addressing these gaps, our work aims to enhance the robustness and efficiency of image transmission systems by focusing on both inference and transmission waiting times, thereby contributing to more reliable and low-latency wireless communication solutions.

Building upon these advancements, we propose a novel adaptive and progressive image transmission pipeline based on state-of-the-art LIC architectures. Specifically, we leverage the strengths of the hyperprior model and VQGAN to address the challenges of dynamic wireless channels by balancing reliability, throughput, and latency.

The hyperprior model is known for its exceptional compression performance due to effective quantization and lossless compression in the bottleneck of the autoencoder. However, it is highly sensitive to bit errors, which limits its application in noisy channels unless robust error correction or retransmission mechanisms are employed. Conversely, VQGAN utilizes vector quantization in the latent space, inherently providing robustness to bit errors and allowing the decoder to reconstruct images even without channel coding. This characteristic enhances reliability in noisy environments but may not achieve the same compression efficiency as the hyperprior model.

To harness the advantages of both models and overcome their individual limitations, we introduce progressive versions of these architectures. Our progressive transmission framework allows for partial image transmission and decoding, enabling immediate availability of coarse images under suboptimal channel conditions or limited throughput. As channel conditions improve or more bandwidth becomes available, additional data can be transmitted to progressively refine the image quality. This approach not only maintains image integrity under poor channel conditions but also significantly reduces latency by allowing immediate partial image availability.

We evaluate our proposed pipeline on the Kodak high-resolution image dataset, measuring performance in terms of peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). Experimental results demonstrate that our progressive transmission framework significantly enhances reliability and reduces latency compared to non-progressive counterparts. Moreover, the progressive approach is particularly beneficial for IoT applications and task-oriented communications, where initial low-quality images are sufficient for immediate processing, and subsequent refinements improve performance as conditions permit.

By integrating adaptive and progressive LIC architectures, this work contributes to the advancement of intelligent communications by providing a robust and efficient solution for real-time image transmission in wireless systems. Our approach addresses the critical need for reliable, low-latency image delivery in environments with dynamic and challenging channel conditions, thereby supporting the demands of real-time computer vision tasks and other emerging applications.

In this work, we present the following key contributions to the field of efficient and robust image transmission over wireless channels:

1. **First Progressive Transmission Framework Based on LIC:** We introduce a novel progressive transmission pipeline tailored for hyperprior-based LIC architectures. To the best of our knowledge, this is the first approach that leverages LIC for progressive image transmission, enabling efficient and adaptable transmission under varying channel conditions.

2. **Use of Residual Vector Quantization with VQGAN:** This work is the first to employ residual vector quantization within a VQGAN-based framework for progressive image transmission. Our method ensures reliable image reconstruction in noisy channels without relying on traditional channel coding. Additionally, the progressive strategy facilitates immediate partial image availability, significantly reducing latency for real-time applications.

3. **Analysis of Waiting Time and Practical Implementation:** We conduct an analysis of waiting time in progressive image transmission, addressing an aspect that previous works have overlooked by focusing primarily on inference time. This analysis provides deeper insights into the real-time performance and responsiveness of our transmission framework. Furthermore, by utilizing readily available image encoders, we enhance the reproducibility and ease of further development of our work, making it more accessible for future research and practical applications.

These contributions collectively advance the state-of-the-art in intelligent image transmission over wireless channels, providing robust, low-latency, and efficient solutions tailored to the demands of modern and emerging applications[1].

7.2 Related Work

7.2.1 Traditional Image Compression Methods

Traditional image compression techniques have been the cornerstone of digital image transmission for decades. Standards such as JPEG [14], WebP [4], and

[1] Project code: https://github.com/M0574F4/LIC_TX

high efficiency video coding (HEVC) [15] are widely adopted due to their balance between compression efficiency and computational complexity. JPEG, one of the earliest compression standards, employs the discrete cosine transform (DCT) to reduce spatial redundancy, followed by quantization and entropy coding [14]. Despite its widespread use, JPEG suffers significant artifacts and quality degradation at extremely low bitrates due to large quantization steps, which discard essential image details [16].

WebP, developed by Google, extends the capabilities of JPEG by incorporating both lossy and lossless compression techniques, leveraging predictive coding and entropy encoding to achieve better compression ratios [4]. HEVC further improves compression efficiency by using advanced techniques such as block partitioning, motion compensation, and in-loop filtering, making it suitable for high-resolution and high-dynamic-range images [15]. However, these traditional methods are highly sensitive to bit errors, especially in noisy wireless channels. Minor bit errors can lead to significant artifacts or render the entire image undecodable, necessitating the use of robust error correction or retransmission strategies to maintain image integrity [6].

7.2.2 Learned Image Compression Methods

Recent advancements in deep learning have encouraged the development of LIC models, which leverage neural networks to outperform traditional compression standards in both compression efficiency and adaptability. Hyperprior models, introduced by Ballé et al. [9], utilize a hypernetwork to model the distribution of latent representations, enabling more efficient entropy coding and improved compression performance. These models achieve state-of-the-art results by optimizing rate-distortion trade-offs through end-to-end training [10].

VQGAN, originally designed as image tokenizers for generative transformers [11], has been repurposed for image compression by leveraging vector quantization in the latent space of autoencoder architectures [7]. Unlike traditional LIC models, VQGAN is not inherently a compression model but offers robust image reconstruction capabilities where errors in the tokenized latent representation result in only localized quality degradation rather than making the entire image undecodable. However, VQGAN's primary focus on image generation rather than compression introduces challenges in achieving optimal compression ratios without compromising image quality.

Other notable LIC approaches include variational autoencoders (VAEs) [17] and generative adversarial networks (GANs) [18], which have been tailored for compression by incorporating probabilistic models and adversarial training to enhance perceptual quality. These models demonstrate significant improvements over traditional methods, particularly in maintaining image fidelity at lower bitrates.

7.2.3 Progressive Transmission in Wireless Communications

Progressive transmission techniques have gained traction as a means to enhance image transmission reliability and reduce latency in wireless communications. Progressive compression schemes, such as scalable video coding (SVC) [19] and layered image compression [20], enable the transmission of image data in multiple layers, allowing receivers to reconstruct low-quality images quickly and progressively enhance them as more data becomes available. This approach is particularly beneficial in scenarios with fluctuating channel conditions, as it ensures that partial data can be utilized effectively without waiting for complete transmission [21].

In the context of wireless communications, progressive transmission aligns well with the need for balancing reliability, throughput, and latency. Techniques such as unequal error protection [22] and adaptive modulation and coding (AMC) [23] have been employed to dynamically adjust transmission parameters based on channel quality. However, integrating progressive transmission with LIC models remains a relatively unexplored area, presenting opportunities for enhancing image transmission robustness and efficiency.

To the best of our knowledge, progressive image transmission tailored specifically for learned image compression has not been extensively investigated. However, researchers outside the communications domain have explored progressive decoding strategies. For instance, [24] developed a progressive decoder, demonstrating the feasibility of controllable rates in an autoencoder with a hyperprior branch. Additionally, Flowers et al. [25] employed a vector quantized variational autoencoder (VQVAE) model and proposed a hierarchical architecture that performs residual vector quantization in the bottleneck of the image autoencoder. These studies highlight the potential of progressive decoding and hierarchical quantization approaches, inspiring our proposed adaptive and progressive image transmission pipeline integrated with LIC models for wireless environments.

7.3 Background and Preliminaries

Efficient image transmission in wireless communications necessitates a comprehensive understanding of both the communication channel characteristics and the fundamentals of image compression techniques. This section provides the essential background required to appreciate the challenges and innovations presented in this study.

7.3.1 System Model

The performance of image transmission systems in wireless environments is intrinsically linked to the properties of the communication channels. To design robust

image compression and transmission schemes, it is crucial to model these channels accurately while maintaining a balance between complexity and practicality.

7.3.1.1 Wireless Channel Model

A wireless communication channel can be mathematically modeled as:

$$y = hx + n, \tag{7.1}$$

where:

- $y \in \mathbb{C}^L$ is the received signal vector,
- $h \in \mathbb{C}^L$ represents the channel coefficients,
- $x \in \mathbb{C}^L$ is the transmitted signal vector,
- $n \sim \mathcal{CN}(0, \sigma^2 I)$ is the AWGN vector with zero mean and covariance matrix $\sigma^2 I$.

and L is dimensionality of the signal vectors. In this model, h encompasses both large-scale fading (path loss, shadowing) and small-scale fading (multipath effects). The AWGN component n captures the thermal noise inherent in the communication system.

This formulation allows us to abstract the complex channel characteristics into an effective transmission rate R, which is crucial for designing adaptive and progressive transmission schemes.

7.3.1.2 Capacity Analysis under Finite Block Length Constraints

Shannon's capacity theorem provides a fundamental limit on the maximum achievable transmission rate C for a given channel, ensuring reliable communication. However, Shannon's theorem assumes infinitely long codewords, which is impractical for real-world applications where finite block lengths n are used. The finite block length capacity introduces a trade-off between throughput and delay:

$$R = C - \sqrt{\frac{V}{n}} Q^{-1}(\epsilon), \tag{7.2}$$

where:

- R is the achievable rate,
- V is the channel dispersion,
- n is the block length,
- $Q^{-1}(\epsilon)$ is the inverse of the Q-function evaluated at the error probability ϵ.

This relationship highlights that shorter block lengths, which are desirable for low latency, result in rates R that are below the Shannon capacity C. Consequently, there is a trade-off between maximizing throughput and minimizing delay, which is particularly relevant for designing progressive and adaptive transmission schemes where latency is a critical factor [26].

For the purposes of this study, we use achievable transmission rate R, which is a function of the available bandwidth B and the signal-to-noise ratio (SNR). According to Shannon's capacity theorem [27], the channel capacity C in bits per second (bps) is given by:

$$C = B \log_2 \left(1 + \frac{P}{N_0 B}\right), \tag{7.3}$$

where:

- P is the transmit power,
- N_0 is the noise power spectral density.

7.3.1.3 Performance Criteria

We consider three criteria based on system parameters and channel conditions: throughput, reliability, and latency, which we will describe in full detail next.

Throughput Throughput refers to the effective data transmission rate and is constrained by the channel capacity C. Typically, in communication systems, throughput is measured in bits per second (bps) to indicate how much data can be transmitted within a given time frame. However, in this work, which focuses on computer vision tasks, we define throughput as pixels per second, as this metric more directly relates to the amount of visual data processed over time. Higher throughput, measured in pixels per second, enables faster image processing and transmission, which is essential for applications requiring real-time or near-real-time data delivery.

Reliability Reliability pertains to the accuracy and integrity of the transmitted image data. It is influenced by factors such as the Bit Error Rate (BER). In task-oriented communications, reliability is defined not only by the BER but also by the successful completion of specific tasks, such as object detection or recognition, based on the received images. However, in this work, where specific tasks are not the primary focus, reliability is primarily assessed using metrics like PSNR and SSIM. These metrics provide a quantifiable measure of image fidelity and integrity, effectively capturing reliability in the absence of task-specific evaluations.

Latency Latency L is the time delay between the initiation of image transmission and its successful reception and reconstruction at the receiver. End-to-end latency can be decomposed into several components:

$$L = L_c + L_t + L_d, \tag{7.4}$$

where:

- L_c is the compression latency,
- L_t is the transmission latency,
- L_d is the decompression latency.

Compression latency (L_c) is the time taken to compress the image before transmission. Transmission latency (L_t) is the time required to transmit the compressed data over the wireless channel, which is influenced by the available transmission rate and the size of the data. Decompression latency (L_d) is the time taken to decompress and reconstruct the image at the receiver.

Progressive transmission primarily aims to reduce the transmission latency (L_t) by enabling the receiver to reconstruct a low-quality version of the image quickly, followed by incremental enhancements as more data becomes available. This approach minimizes the perceived latency, allowing for immediate partial image availability at the cost of initially lower image quality.

7.3.2 Learned Image Compression Fundamentals

LIC leverages deep learning techniques to surpass traditional compression methods in terms of compression efficiency and adaptability to varying channel conditions. LIC models typically employ neural network architectures that are trained end-to-end to optimize the trade-off between compression rate and image quality.

7.3.2.1 Overview of Autoencoder Architectures

At the heart of most LIC models lies the autoencoder architecture, which comprises an encoder, a bottleneck (latent space), and a decoder [9]. The encoder transforms the input image I into a compact latent representation z, which is then quantized and compressed for transmission. The decoder reconstructs the image $\hat{I}$ from the compressed latent code:

$$z = \text{Quantize}(\text{Encoder}(I)), \tag{7.5}$$

$$\hat{I} = \text{Decoder}(z). \tag{7.6}$$

The autoencoder framework is highly flexible and can be enhanced with various mechanisms to improve compression efficiency and image quality. For instance, incorporating attention mechanisms or residual connections can enhance the model's

ability to capture intricate image details [10]. Additionally, probabilistic models within the autoencoder allow for better entropy modeling, leading to more efficient compression [28].

7.3.2.2 Vector Quantization in Latent Spaces

Vector quantization (VQ) is a technique used to discretize the continuous latent representations produced by the encoder. In the context of LIC, VQ reduces the dimensionality of the latent space, facilitating more efficient compression [29]. For example, VQGAN utilize VQ in their autoencoder architectures to create discrete latent codes that are easier to compress and transmit [11].

The integration of VQ in LIC models without lossless compression offers several advantages. Discrete latent spaces are inherently more resilient to transmission noise, enhancing the reliability of image reconstruction [30]. Additionally, hierarchical VQ approaches, such as residual vector quantization, enable the progressive refinement of image details by encoding residual information at multiple levels [25]. This hierarchical structure not only improves compression efficiency but also supports progressive transmission by allowing partial data to enhance image quality incrementally. Furthermore, recent advancements have explored the use of transformer architectures in conjunction with VQ-based models for generating high-fidelity images from discrete latent codes, bridging the gap between compression and image generation [11].

These features make VQ-based LIC models particularly well-suited for adaptive and progressive image transmission in dynamic wireless environments.

7.4 Proposed Methodology

This section presents our novel adaptive and progressive image transmission pipeline tailored for learned image compression (LIC)-based architectures. We explore two state-of-the-art LIC models: a hyperprior-based model and a VQGAN. Our proposed pipeline is designed to operate efficiently under dynamic channel conditions, balancing reliability, throughput, and latency.

7.4.1 System Architecture Overview

Our transmission pipeline is structured around two primary LIC models: the hyperprior-based model and the VQGAN-based model. Each model's architecture is illustrated in figs. 7.1 and 7.2.

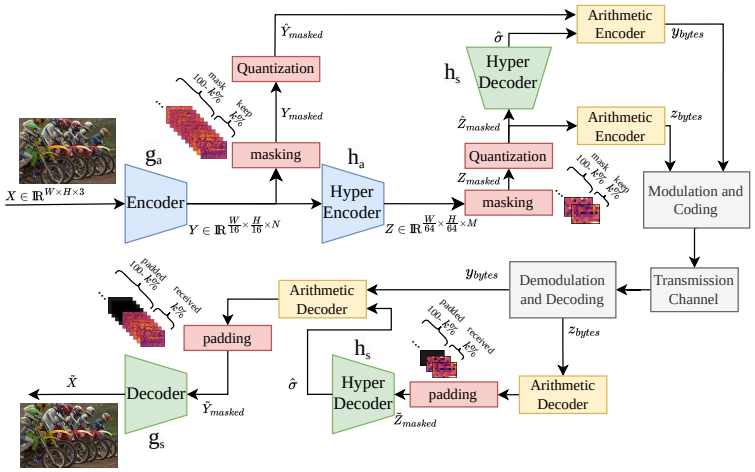


Figure 7.1: System model for transmission based on the hyperprior model, illustrating the retention of $k\%$ of feature maps while masking the remainder in Y and Z at the transmitter, and the reconstruction by padding zeroes in the masked $100-k\%$ feature maps at the receiver. The selection of k is based on channel conditions and service requirements.

7.4.1.1 Hyperprior-Based Model Integration

The hyperprior-based model utilizes a variational autoencoder structure that enhances compression efficiency through the use of a hyperprior. The architecture of this model, depicted in fig. 7.1, employs the following mathematical formulation [9]:

$$Y = g_a(X), \tag{7.7}$$
$$Y_{\text{masked}} = \text{Mask}(Y, k\%), \tag{7.8}$$
$$\hat{Y}_{\text{masked}} = \text{Quantize}(Y_{\text{masked}}), \tag{7.9}$$
$$Z = h_a(Y), \tag{7.10}$$
$$Z_{\text{masked}} = \text{Mask}(Z, k\%), \tag{7.11}$$
$$\hat{Z}_{\text{masked}} = \text{Quantize}(Z_{\text{masked}}), \tag{7.12}$$
$$\hat{\sigma} = h_s(\hat{Z}_{\text{masked}}), \tag{7.13}$$
$$y_{\text{bytes}} = \text{ArithmeticEncode}(\hat{Y}_{\text{masked}}, \hat{\sigma}), \tag{7.14}$$
$$z_{\text{bytes}} = \text{ArithmeticEncode}(\hat{Z}_{\text{masked}}), \tag{7.15}$$

where,

- g_a is the encoder that transforms the input image $X \in \mathbb{R}^{W\times H\times 3}$ into a latent representation $Y \in \mathbb{R}^{\frac{W}{16}\times\frac{H}{16}\times N}$.

- Mask$(Y, k\%)$ masks $k\%$ of the channels in Y, preserving the remaining channels.
- Quantize(Y_{masked}) quantizes the masked latent representation to produce $\hat{Y}_{\text{masked}}$.
- h_a is the hyperprior encoder that processes Y to generate hyper latent $Z \in \mathbb{R}^{\frac{W}{64} \times \frac{H}{64} \times M}$.
- Mask$(Z, k\%)$ and Quantize(Z_{masked}) follow similar processing steps for the hyper latent.
- h_s decodes $\hat{Z}_{\text{masked}}$ to estimate the scale parameter $\hat{\sigma}$, crucial for arithmetic encoding.
- ArithmeticEncode generates byte streams y_{bytes} and z_{bytes} using the quantized and masked representations.

At the receiver end, depicted in fig. 7.1, the decoding process unfolds as follows:

$$\tilde{Z}_{\text{masked}} = \text{ArithmeticDecode}(z_{\text{bytes}}), \tag{7.16}$$
$$\hat{\sigma} = h_s(\tilde{Z}_{\text{masked}}), \tag{7.17}$$
$$\tilde{Y}_{\text{masked}} = \text{ArithmeticDecode}(y_{\text{bytes}}, \hat{\sigma}), \tag{7.18}$$
$$\tilde{X} = g_s(\tilde{Y}_{\text{masked}}), \tag{7.19}$$

where:

- g_s is the decoder that reconstructs the image $\tilde{X}$ from the processed $\tilde{Y}_{\text{masked}}$.

During training, a random selection of $k\%$ of the feature maps is masked to train the network to focus on significant features where $k = 100 \times u$ and $u \sim \text{Uniform}(0,1)$. During inference, the selection of $k\%$ is dynamically adjusted based on real-time channel conditions and application requirements, enabling adaptive transmission efficiency. This adaptive selection allows the system to decode the image from a minimal number of feature maps initially, with subsequent transmissions providing additional feature maps to enhance image quality progressively.

7.4.1.2 VQGAN-Based Model Integration

The VQGAN-based model employs vector quantization within its generative adversarial network architecture. This model's system architecture is illustrated in fig. 7.2, with the following process flow:

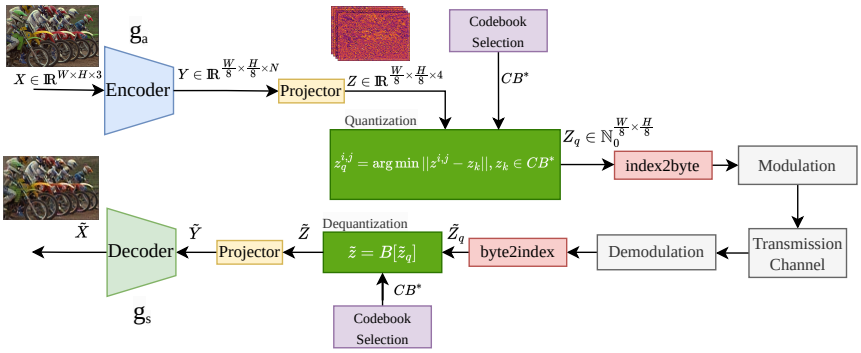


Figure 7.2: System model for VQGAN-based image transmission, illustrating the generation of feature maps (Y) and token maps (Z) during the encoding and decoding process.

$$Y = g_a(X), \tag{7.20}$$
$$Z = P_E(Y), \tag{7.21}$$
$$Z_q = \text{VectorQuantize}(Z, CB), \tag{7.22}$$
$$\tilde{Z} = \text{Dequantize}(\tilde{Z}_q, CB), \tag{7.23}$$
$$\tilde{Y} = P_D(\tilde{Z}), \tag{7.24}$$
$$\tilde{X} = g_s(\tilde{Y}), \tag{7.25}$$

where,

- g_a is the encoder transforming the input image X into the latent representation $Y \in \mathbb{R}^{\frac{W}{8} \times \frac{H}{8} \times N}$.
- P_E projects Y into a lower-dimensional space $Z \in \mathbb{R}^{\frac{W}{8} \times \frac{H}{8} \times 4}$.
- VectorQuantize(Z, CB) quantizes Z into the token map Z_q, using the codebook CB.

Additionally, the VQGAN-based model incorporates residual codebook clustering, enabling a hierarchical architecture that performs residual vector quantization in the bottleneck of the image autoencoder. This hierarchical structure allows for progressive decoding by transmitting indices corresponding to coarse and fine codebooks in successive transmissions, thereby refining the image quality incrementally without relying on lossless compression.

7.4.2 Adaptive and Progressive Transmission Mechanism

Our progressive transmission mechanism is designed to adaptively adjust the amount of data transmitted based on real-time channel conditions and application-specific requirements. This mechanism is integral to both the hyperprior-based

and VQGAN-based models, enabling them to balance reliability, throughput, and latency effectively.

7.4.2.1 Adaptive Pipeline Design

Rather than redesigning source and channel coding [6, 12, 13], our approach leverages existing systems where modulation and coding schemes are dynamically selected based on the channel condition. This adaptability allows our pipeline to be seamlessly integrated into various communication infrastructures.

In each transmission slot, a permissible bit budget N_{bits}, determined by the current channel rate—which is influenced by factors such as bandwidth and noise level—is allocated. Based on this N_{bits}, the system performs the following selections:

Hyperprior-Based Model: Selects the maximum number of feature maps such that the top $k\%$ of channels fit within the allocated bit budget.

VQGAN-Based Model: Chooses the number of relevant codebooks for vector quantization based on the available channel rate.

7.4.2.2 Progressive Image Decoding

Hyperprior-Based Model In this experiment, we analyzed the sensitivity of the Learned Image Compression-TCM (LIC-TCM) model from Liu et al. [10] to channel masking. Specifically, we initiated the masking of a percentage of channels in the bottleneck and observed non-uniform sensitivity across different channels. Masking certain channels resulted in significant PSNR degradation, while others had minimal impact.

To further investigate, we treated the problem as a feature pruning task. We performed inference on 100 randomly selected batches of images (batch size = 8) from the ImageNet dataset, systematically masking each feature map and recording the Mean Squared Error (MSE) degradation in image reconstruction. By averaging the MSE degradation across all tested images, we derived an importance metric for each feature map.

Our results demonstrated that masking channels in order of least to most importance based on the averaged MSE degradation achieved a smooth PSNR drop and maintained better image reconstruction quality across varying mask percentages, as illustrated in fig. 7.3. This finding motivates our progressive image transmission approach for hyperprior-based models. We utilize the sorting mechanism from Hojjat et al. [24], which ranks feature maps according to their importance metrics, allowing for an adaptive and progressive transmission strategy that prioritizes critical feature maps under varying channel conditions.

At the receiver, the progressive transmission is handled by maintaining the previously received $\tilde{Y}_{\text{masked}}$ feature maps. With each new transmission, additional feature

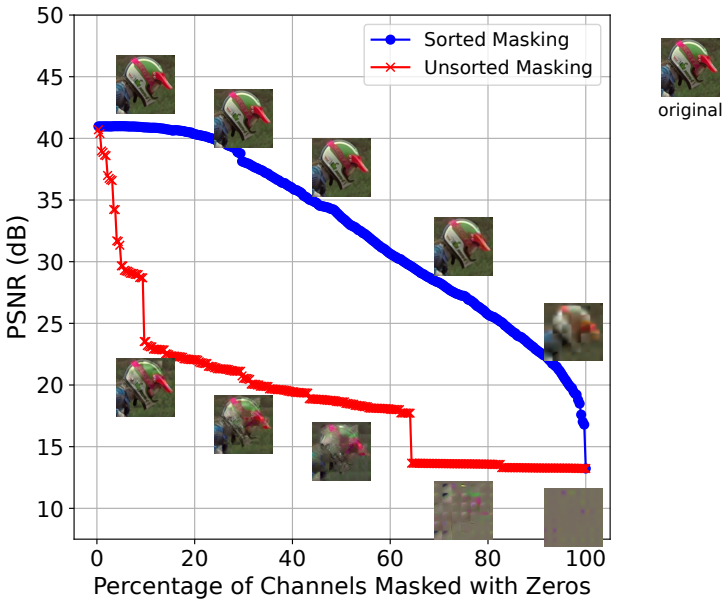


Figure 7.3: Decrease in Peak Signal-to-Noise Ratio (PSNR) versus the percentage of masked bottleneck channels for the image kodim05. The red line represents unsorted masking, while the blue line corresponds to observer-based (sorted) feature masking.

maps are received and concatenated with the existing ones, allowing the decoder to incrementally refine the reconstructed image $\tilde{X}$. This approach ensures that a low-quality version of the image is available quickly, with subsequent transmissions enhancing the image quality as more data becomes available.

VQGAN-Based Model For the VQGAN-based model, we adopt an advanced hierarchical vector quantization approach to facilitate progressive image transmission. Building upon the methodology proposed by Zhu et al. [31], we use their projector to compress the embedding dimension to 4, thereby reducing the latent space complexity. Subsequently, it expands the codebook size to 100K, establishing a large codebook that serves as the foundation for our residual vector quantization-based progressive transmission.

To manage this extensive codebook efficiently, we perform two types of clustering:

K-Means-Based Clustering: We utilize Facebook AI Similarity Search (Faiss) [32] to perform k-means clustering, partitioning the large codebook into smaller, manageable subsets corresponding to different bits per index (bpi). The clustering is configured with a variable number of clusters, specifically 2^{bpi}, to represent different reconstruction qualities. For each desired bpi, the algorithm clusters the

embeddings into 2^{bpi} clusters. These clustered codebooks are then extracted and stored, facilitating efficient codebook selection based on the available bit budget N_{bits} during transmission.

The codebook selection process is formalized as an optimization problem:

$$CB^* = \arg\max_{B} \quad bpi \quad \text{s.t.} \quad n_{\text{bits}} \leq N_{\text{bits}}, \tag{7.26}$$

where the number of bits n_{bits} is defined as:

$$n_{\text{bits}} = \left(\frac{W}{8}\right)\left(\frac{H}{8}\right) bpi, \tag{7.27}$$

with B^* being the selected codebook among all available codebooks, CB, $W \times H$ denoting the dimensions of the image, and 8 being the down sampling factor across each spatial dimension of the encoder. Here, bpi represents the number of bits per index corresponding to the selected codebook. This formulation ensures that the selected codebook maximizes the bits per index without exceeding the transmission bit budget.

Residual Quantizer (RQ): For progressive transmission, we implement a residual quantizer (RQ) that quantizes the residuals of the input vectors sequentially. At each encoding stage m, the RQ selects the codeword c_m that best approximates the residual of the input vector x relative to the previously encoded steps:

$$c_m = \underset{j=1,\ldots,K}{\text{argmin}} \left\| \sum_{i=1}^{m-1} T_i[c_i] + T_m[j] - x \right\|^2, \tag{7.28}$$

where $T_i[c_i]$ represents the codeword from the i^{th} transmission stage indexed by c_i, and K is the number of codewords in the codebook. This sequential encoding allows the receiver to reconstruct the image progressively by first decoding coarse details and then refining them with finer residuals as more indices are received.

7.4.2.3 Progressive Transmission Strategy

In the VQGAN-based model, the receiver maintains a list of previously decoded indices corresponding to coarse codebooks. As new indices referencing finer codebooks are received, they are integrated with the existing indices to progressively decode and enhance the image. This hierarchical decoding process enables the immediate reconstruction of a basic image from the coarse indices, followed by incremental improvements as additional fine-grained indices are received.

To optimize progressive transmission, we dynamically determine the number of encoding stages that can be transmitted within the available bit budget N_{bits}. Given that each stage of residual quantization requires a fixed number of bits n_{bits}, the maximum number of stages M_{stages} that can be accommodated is calculated as:

$$M_{\text{stages}} = \min\left(\left\lfloor \frac{N_{\text{bits}}}{n_{\text{bits}}} \right\rfloor, M_{\text{stages}}^{\text{max}}\right), \tag{7.29}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation and $M_{\text{stages}}^{\max}$ is the maximum number of codebooks learned for the residual vector quantization. This ensures that the total number of bits used does not exceed the budget N_{bits}.

Each encoding stage corresponds to a residual codebook that quantizes the residuals of the input vector relative to the previous stages. All residual codebooks are of the same size, and each stage incrementally refines the image quality by addressing the residual errors from earlier stages. By determining M_{stages}, we can transmit up to M_{stages} stages of residual quantization within the given bit budget, allowing the receiver to progressively enhance the image quality as more data becomes available.

This adaptive selection of encoding stages based on the bit budget ensures that the most critical residuals are transmitted first, facilitating a balanced trade-off between image quality and transmission efficiency. Consequently, our approach leverages residual vector quantization to enhance the fidelity of the reconstructed image progressively, enabling robust and scalable image transmission under varying channel conditions without relying on lossless compression.

Using adaptive and progressive image transmission pipelines that integrate hyperprior-based and VQGAN-based LIC models to address the challenges of dynamic wireless channels by dynamically adjusting the transmitted data based on real-time channel conditions and employing progressive decoding strategies, we effectively balance reliability, throughput, and latency.

7.5 Experimental Setup

To evaluate the performance of our proposed adaptive and progressive image transmission pipeline, we conducted comprehensive experiments designed to assess the efficacy of our models under realistic wireless communication conditions. This section details the experimental setup, including the datasets used, model configurations, simulation parameters, implementation environment, and evaluation metrics.

7.5.1 Dataset and Models

7.5.1.1 Dataset

We utilized the widely recognized Kodak dataset [33] for evaluating our models. Kodak Dataset Consists of 24 high-quality, uncompressed color images frequently used as a benchmark for image compression and transmission evaluations. The images cover a variety of scenes and are of size 768×512 or 512×768 pixels.

All images were preprocessed to match the input requirements of the pretrained models. No data augmentation techniques were applied during testing to maintain consistency.

Table 7.1: Notations

Symbol	Description
R	Achievable transmission rate (bps)
B	Available bandwidth (Hz)
SNR	Signal-to-noise ratio
C	Channel capacity (bps)
P	Transmit power (W)
N_0	Noise power spectral density (W/Hz)
L	Latency (seconds)
L_c	Compression latency (seconds)
L_t	Transmission latency (seconds)
L_d	Decompression latency (seconds)
BER	Bit error rate
PSNR	Peak signal-to-noise ratio
SSIM	Structural similarity index measure
$h \in \mathbb{C}^L$	Channel coefficients
X	input image
Y	Latent representation of X
Y_{masked}	Masked latent representation Y
$\hat{Y}_{\text{masked}}$	Quantized masked latent representation
Z	Hyper latent
Z_{masked}	Masked hyper latent Z
$\hat{Z}_{\text{masked}}$	Quantized masked hyper latent Z
$\hat{\sigma}$	Estimated scale parameter
$\tilde{Z}_{\text{masked}}$	Decoded masked hyper latent Z
$\tilde{Y}_{\text{masked}}$	Decoded masked latent representation Y
$\tilde{X}$	Reconstructed image
CB^*	Selected codebook
$M_{\text{stages}}^{\text{max}}$	Maximum number of codebooks in RQ

7.5.1.2 Models

Our study investigates two primary LIC models: a hyperprior-based model and a VQGAN-based model.

Hyperprior-Based Model We employed the architecture described in [24], which is based on a hyperprior variational autoencoder for image compression. The model configurations are as follows:

- Latent Channels (N_Y): 192 channels
- Hyper Latent Channels (N_Z): 128 channels

and is trained using rate-distortion parameter $\lambda = 0.1$, and for using 0-100$\%$ of the channels. This particular model is

VQGAN-Based Model For the VQGAN-based model, we utilized the LIC architecture proposed in [31], which integrates vector quantization with GANs for image compression. The model specifications include:

- Codebook: A large initial codebook of 100,000 entries.
- Projection: A projector that compresses the embedding dimension to 4, facilitating efficient vector quantization.

We use pretrained models on the ImageNet dataset [34], and no additional training was conducted as part of this study.

Adaptive WebP Model In addition to the learned image compression models, we utilize adaptive WebP [4] as a benchmark. Adaptive WebP is a widely adopted, non-learning-based compression standard that can be used to dynamically adjust compression ratio based on channel conditions. We chose Adaptive WebP as a benchmark because similar to our progressive transmission approach it can be used to optimize performance by adjusting compression settings in real-time. Furthermore, unlike some learning-based image transmission pipelines that are not publicly available, Adaptive WebP is accessible to the public, ensuring transparency and reproducibility in our comparative analysis.

7.5.2 Implementation Environment

7.5.2.1 Hardware

All experiments were conducted on a high-performance computing setup equipped with:

- Processor: Intel(R) Xeon(R) Silver 4310 CPU @ 2.10GHz with 48 cores.
- Graphics Processing Unit (GPU): NVIDIA A40 with 46,068 MiB memory, supporting accelerated computations for deep learning models.

7.5.2.2 Software

The software environment was configured as follows:

- Operating System: Ubuntu 20.04 LTS.
- Deep Learning Framework: PyTorch version 2.2.1.

- CUDA: Version 12.2, enabling GPU acceleration.
- FAISS [32]: Utilized for efficient k-means clustering during codebook generation.
- Additional Libraries: NumPy, SciPy, and other standard scientific computing libraries.

7.5.3 Channel Simulation

To model realistic wireless communication environments characterized by multipath propagation and rapid signal attenuation, we employed a Rayleigh fading channel model. This model is widely used to represent multipath fading in wireless communications. The channel coefficients were generated using the sum-of-sinusoids method, approximating Clarke's model for flat fading channels [35]. The key parameters of the channel simulation are as follows:

- Maximum Doppler Frequency (f_d): 10 Hz, representing a moderate level of mobility.
- Symbol Duration (T_s): 1 ms.
- Bandwidth (B): 100 kHz.
- SNR: Varied over $\{$-10, -5, 0, 5$\}$ dB to simulate different channel conditions.

We are mainly interested in image transmission in challenging channel conditions hence low SNRs and limited channel bandwidth are assumed. To ensure statistical reliability and average out the randomness inherent in Rayleigh fading channels, we simulated 1,000 independent channel realizations.

7.5.4 Performance Metrics

Evaluating the performance of image compression and transmission systems involves multiple metrics that assess various aspects of system efficiency and image quality. The key metrics considered in this study are PSNR, SSIM, throughput, and latency.

7.5.4.1 Image Quality Metrics

PSNR is a widely used objective metric for measuring the quality of reconstructed images compared to the original images. It is defined as:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right), \tag{7.30}$$

where MAX represents the maximum possible pixel value of the image and Mean Squared Error (MSE) is given by:

$$\text{MSE} = \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \left(I(i,j) - \hat{I}(i,j) \right)^2, \tag{7.31}$$

with $N_1 \times N_2$ being the dimensions of the image.

The SSIM evaluates the similarity between two images based on luminance, contrast, and structural information. It is calculated as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{7.32}$$

where μ_x and μ_y are the mean values, σ_x^2 and σ_y^2 are the variances, and σ_{xy} is the covariance of the original and reconstructed images. C_1 and C_2 are constants to stabilize the division. SSIM values range from -1 to 1, with higher values indicating greater structural similarity between images.

7.5.4.2 Transmission Performance Metrics

Latency is defined as the time delay between the initiation of image transmission and its successful reception and reconstruction at the receiver. However, in our experiments, latency was quantified by the number of transmission slots required to transmit the image, excluding encoding and decoding times. We further break down latency metrics into:

- Average Waiting Time (T_{avg}): The mean time delay (in ms) for image transmission.
- 99.9th Percentile Waiting Time ($T_{99.9\%}$): The time below which 99.9% of image transmissions are completed, critical for delay-sensitive applications.

Throughput is measured as the effective data rate achieved during transmission, influenced by the modulation scheme and channel conditions. We use Megapixels per second (Mpps) as the unit, where each pixel is counted only once to reflect the actual image area transmitted. This metric ensures that multiple bits transmitted to refine the same pixels do not inflate the throughput measurement.

Reliability assesses the accuracy and integrity of the transmitted image data. In this study, reliability is evaluated based on the fidelity of image reconstruction as indicated by PSNR and SSIM values under varying channel conditions. Additionally, reliability encompasses the successful completion of image transmission without significant degradation, especially under poor SNR conditions.

7.5.5 Implementation Details

7.5.5.1 VQGAN-based model

For the VQGAN-based model, codebook generation and clustering are critical for progressive transmission. K-means clustering was employed using FAISS [32] for efficient clustering on embeddings. Codebooks were generated for bits per index (bpi) ranging from 8 to 16 with a maximum of 100 iterations for the clustering algorithm and Euclidean distance (L2 norm) as the distance metric. Ultimately, $M_{\text{stages}} = 10$ residual codebooks were trained, each with $bpi = 8$ (256 codewords), to support progressive residual transmission in the VQGAN-based model. For the VQGAN-based model, progressive transmission is achieved by transmitting indices corresponding to coarse codebooks first, followed by finer codebooks. The receiver integrates these indices to incrementally refine the image reconstruction.

7.5.5.2 Progressive Transmission Implementation

In the hyperprior-based model, we implemented progressive transmission by masking feature maps based on their importance, as detailed in Section IV. The receiver progressively reconstructs the image as more feature maps are received.

7.5.6 Experimental Procedure

The experimental workflow was designed to systematically evaluate both the hyperprior-based and VQGAN-based models under identical channel conditions. The procedure is as follows:

1. Data Preparation: The Kodak dataset images were preprocessed to match the input requirements of the models.
2. Codebook Generation: For the VQGAN-based model, codebooks were generated using k-means clustering and residual quantization as described above.
3. Channel Simulation: Simulated Rayleigh fading channels with varying SNR levels were generated for transmission simulations.
4. Transmission Simulation: Each image was subjected to transmission over the simulated channels. The bit budget N_{bits} for each transmission slot was determined based on the current SNR and selected modulation scheme.
5. Progressive Decoding: The receiver progressively reconstructed images by integrating newly received data with previously received data, allowing for incremental improvements in image quality.
6. Metric Collection: At each stage of progressive decoding, PSNR, SSIM, latency, and throughput metrics were recorded to evaluate performance.

7. Result Aggregation: Results were averaged over all images in the dataset and multiple channel realizations to obtain statistically significant performance evaluations.

7.6 Results and Discussion

In this section, we present experimental results evaluating the performance of our proposed adaptive and progressive image transmission pipeline over simulated Rayleigh fading channels. We analyze image quality, latency, throughput, and the effectiveness of our adaptive strategies. We compare the performance of the hyperprior-based and VQGAN-based models under various channel conditions and bit budgets, and discuss the implications of our findings for wireless communications.

7.6.1 Experimental Results

Table 7.2 illustrates the performance of our proposed models across various SNR values, providing a comprehensive view of throughput, latency, and image quality under realistic channel conditions. Each model is configured to maximize its performance within its respective architecture. The progressive-hyperprior model is designed to transmit up to 32 feature maps, balancing between resolution and minimal latency. Progressive-VQGAN operates with 10 residual quantization stages ($M_{\text{stages}} = 10$), aiming for higher fidelity under limited channel conditions. Finally, adaptive WebP dynamically adjusts quality factors (here from 1 to 4) to utilize available channel bandwidth effectively, optimizing image quality.

A snapshot of the Rayleigh fading channel magnitude $|h|$ over a timespan of 300 ms is presented in fig. 7.4. This plot illustrates the temporal variations in channel conditions, which directly impact the image quality and transmission latency of each model. As observed, significant drops in channel magnitude correspond to stopped transmission for adaptive WebP, where it fails to deliver even the lowest quality image (quality factor = 1). In contrast, the progressive transmission schemes continue to function under these challenging channel conditions. This resilience is crucial for applications requiring consistent image quality under varying channel conditions.

7.6.2 Analysis of Results

The experimental data offers insights into each model's strengths in terms of latency, image quality, and throughput across diverse SNR settings.

The Progressive-Hyperprior model consistently outperforms the other models in minimizing latency across all SNR conditions. At low SNR levels (-10, -5, and 0

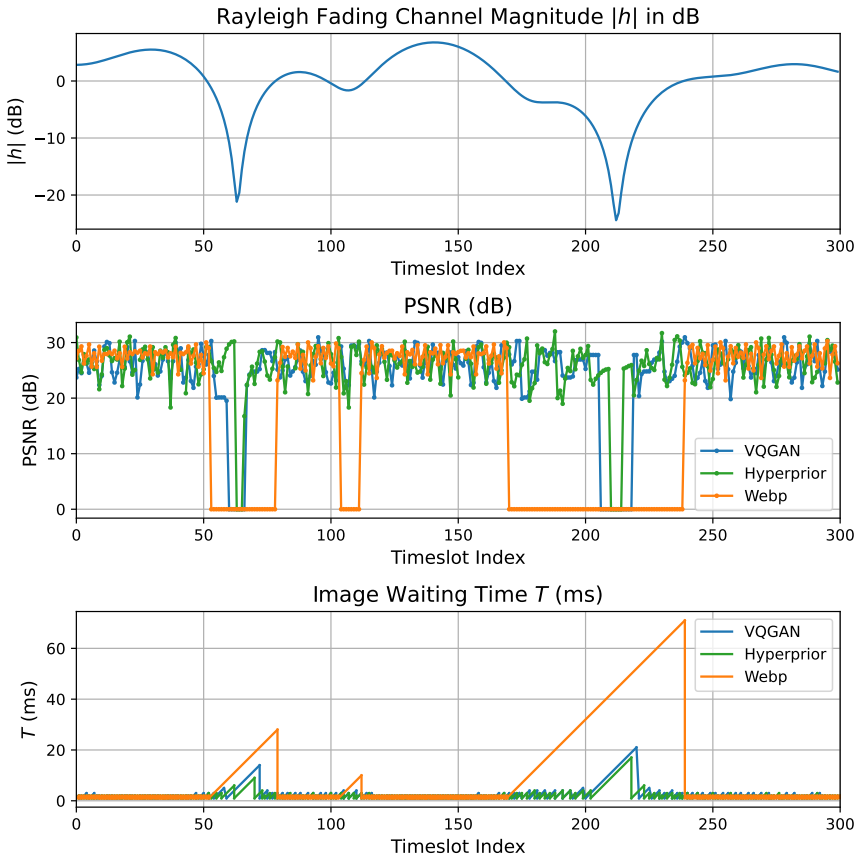


Figure 7.4: A 300 ms snapshot of the fading channel magnitude. The top plot shows the variation in channel magnitude $|h|$ due to fading. The middle and bottom plots respectively present the resulting PSNR and waiting time T for transmitting images using progressive-hyperprior, progressive VQGAN, and adaptive WebP. The figure highlights how each model adapts to channel fluctuations, revealing trade-offs between latency and image quality emphasizing their robustness under dynamic channel conditions.

dB), it achieves the lowest average waiting time T_{avg} and 99.9th percentile waiting time $T_{99.9\%}$, demonstrating its effectiveness for real-time applications where speed is essential. Notably, even at 5 dB, where Adaptive WebP briefly offers a slightly lower T_{avg}, Progressive-Hyperprior maintains a strong lead in 99.9th percentile latency, supporting its robustness under both challenging and favorable conditions for delay-sensitive applications.

This superior latency performance is primarily due to the progressive-hyperprior model's design, which bases its progressive decoding on compact feature maps. These compact feature maps enable the transmission of a granular and small number

Table 7.2: Kodak Test Results for Proposed Models Across SNR Values.

SNR (dB)	Method	Throughput Mpps	PSNR dB	SSIM	T_{avg} ms	$T_{99.9\%}$ ms
-10	Adaptive Webp	0.00	-	-	-	-
	Progressive-VQGAN	4.88	**25.64**	**0.71**	63.14	272.00
	Progressive-Hyperprior	**18.00**	24.99	0.68	**21.52**	**108.00**
-5	Adaptive Webp	0.00	-	-	-	-
	Progressive-VQGAN	34.13	**25.83**	**0.72**	11.14	94.00
	Progressive-Hyperprior	**66.38**	25.28	0.69	**6.62**	**60.00**
0	Adaptive Webp	74.63	**27.47**	**0.75**	5.37	265.00
	Progressive-VQGAN	101.25	26.20	0.73	4.31	34.00
	Progressive-Hyperprior	**160.50**	25.94	0.72	**3.33**	**21.00**
5	Adaptive Webp	**454.50**	**27.63**	**0.76**	**1.82**	71.00
	Progressive-VQGAN	205.13	26.31	0.73	2.67	16.00
	Progressive-Hyperprior	307.88	26.35	0.73	2.22	**12.00**

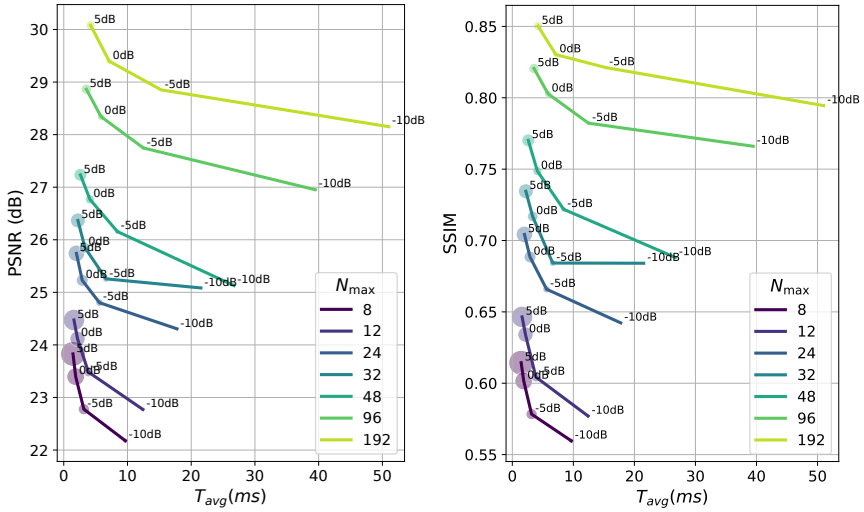
of bits, even in poor channel conditions.

Image quality, as measured by PSNR and SSIM, varies significantly with SNR and model choice: Adaptive WebP yields the highest quality metrics (PSNR and SSIM) at higher SNR levels (0 dB and 5 dB), achieving peak fidelity in ideal channel conditions. However, it cannot transmit at -10, -5, and 0 dB due to channel capacity constraints. In contrast, progressive-VQGAN excels at lower SNR levels, outperforming progressive-hyperprior in terms of PSNR and SSIM. Its architecture is resilient to adverse conditions, providing higher image quality when Adaptive WebP fails to transmit. This suggests progressive-VQGAN's suitability for scenarios demanding quality retention under limited bandwidth.

Throughput performance underscores the progressive-hyperprior's efficiency: At low to moderate SNR (-10 to 0 dB), progressive-hyperprior achieves the highest throughput, indicating optimal use of available channel capacity even under stringent conditions. This throughput advantage makes it suitable for applications where maximizing data transfer is essential despite poor channel quality. At 5 dB, Adaptive WebP surpasses other models in throughput, taking advantage of the higher quality factor options available at improved SNR. This transition illustrates Adaptive WebP's capacity to exploit good channel conditions effectively but also underscores its dependency on sufficient bandwidth.

Fig. 7.5 illustrates the impact of the parameter N_{max} on the tradeoff between throughput, reliability (PSNR and SSIM), and latency for the hyperprior model. The parameter N_{max} specifies the total number of feature maps to be transmitted for each image, enabling us to adapt the model's performance to different channel conditions. By examining this figure, we can select the appropriate value of N_{max}

to achieve a desired tradeoff. For instance, if the target performance requirements are SSIM > 0.75 and PSNR > 27 dB, then $N_{\max} = 96$ and $N_{\max} = 192$ both meet these criteria. However, since $N_{\max} = 96$ allows for higher throughput (transmitting more images per second), it would be the preferred choice. This analysis helps the selection of $N_{\max}$ based on specific performance and throughput priorities.



(a) PSNR (dB) vs. T_{avg} (ms) *(b) SSIM vs. T_{avg} (ms)*

Figure 7.5: Visualization of T_{avg} (ms) and performance metrics for the hyperprior model: (a) PSNR (dB) and (b) SSIM. The size of the circles represents throughput, with annotations on the circles indicating the channel SNR. By adjusting N_{max}, which determines the number of feature maps to be transmitted for each image, we can observe the impact on PSNR and SSIM, reflecting the tradeoff between reliability, latency, and throughput in varying channel conditions.

7.6.3 Discussion

The findings emphasize the adaptability of each model to specific use-case demands, showcasing the trade-offs between latency, quality, and throughput. Progressive-hyperprior is ideal for delay-sensitive applications, minimizing latency while maintaining throughput across various SNR levels. Its selective transmission of feature maps allows it to remain functional under poor conditions, a trait that could benefit applications such as interactive video streaming or remote sensing. Progressive-VQGAN's robustness in low SNR conditions makes it advantageous where quality is a priority and latency is secondary. Its vector quantization approach allows reliable image reconstruction, reducing the need for channel coding (hence encoding and decoding time) lowering computational load. Progressive-VQGAN's perfor-

mance under degraded channels highlights its utility for applications that prioritize visual fidelity under bandwidth constraints. Performing best in high-quality conditions (5 dB), Adaptive WebP achieves the highest image quality and throughput in such environments. However, its dependency on channel capacity restricts its usability in lower SNR scenarios. Adaptive WebP's efficiency in optimal conditions positions it as an ideal choice for high-definition media streaming where channel conditions are controlled or consistently high.

7.6.4 Implications for Wireless Communications

The adaptability and progressive nature of these models provide valuable tools for optimizing wireless communication systems, particularly in variable or constrained environments. By tailoring the transmission approach to the channel's real-time conditions, our framework enables targeted optimization for latency, quality, or throughput as the application requires.

Real-time and low-latency applications can benefit from progressive-hyperprior's latency performance, which supports immediate data access needed for interactive services. For scenarios where image quality is essential, particularly under stable and high SNR conditions, Adaptive WebP or progressive-VQGAN can achieve high PSNR and SSIM values, making them ideal for applications where visual clarity is crucial. The robustness of progressive-VQGAN without channel coding reduces processing demands, beneficial in resource-limited environments or applications where computational efficiency is critical. The robustness of the models is further supported by their ability to maintain image transmission under challenging channel conditions where traditional methods fail, as demonstrated by the continued functionality of the progressive models even when the channel capacity is severely limited.

7.7 Conclusion

This paper introduces an adaptive and progressive image transmission pipeline designed for wireless communication systems utilizing LIC architectures. By integrating hyperprior-based and VQGAN-based models, our approach effectively enhances reliability, throughput, and latency, particularly excelling in challenging wireless channel conditions characterized by low signal-to-noise ratios (SNR).

Notably, the progressive hyperprior model achieves the lowest latency, making it the optimal choice for delay-sensitive applications. Meanwhile, the progressive VQGAN model delivers similar image quality without relying on channel coding, making it highly suitable for scenarios where implementing channel coding is impractical or too costly.

In contrast, non-progressive methods like adaptive WebP exhibit considerable

performance degradation under low SNR conditions, underscoring the robustness and effectiveness of our LIC architectures. These results highlight the substantial benefits of incorporating advanced LIC architectures and learning-based compression techniques into wireless communication protocols.

Our adaptive and progressive image transmission pipeline effectively balances the trade-offs between reliability, throughput, and latency in wireless communication systems. By utilizing advanced LIC architectures and tailoring the transmission strategy to real-time channel conditions, we have demonstrated substantial improvements over traditional methods. The ability to maintain robust image transmission even under challenging conditions opens up new possibilities for a wide range of applications that rely on timely and accurate visual data. The integration of learning-based compression techniques into wireless communication protocols represents a significant step toward meeting the demands of next-generation communication networks.

References

[1] C. Liang, H. Du, Y. Sun, D. Niyato, J. Kang, D. Zhao, and M. A. Imran. *Generative AI-driven Semantic Communication Networks: Architecture, Technologies and Applications*. IEEE Transactions on Cognitive Communications and Networking, pages 1–1, 2024. doi:10.1109/TCCN.2024.3435524.

[2] J. Li and X. Hou. *Exploiting Inter-Image Similarity Prior for Low-Bitrate Remote Sensing Image Compression*. arXiv preprint arXiv:2407.12295, 2024.

[3] A. Skodras, C. Christopoulos, and T. Ebrahimi. *The JPEG 2000 still image compression standard*. IEEE Signal Processing Magazine, 18(5):36–58, 2001. doi:10.1109/79.952804.

[4] Google Developers. *WebP*. https://developers.google.com/speed/webp/docs/compression. Accessed: 2023-02-14.

[5] Fabrice Bellard. *BPG Image Format*. https://bellard.org/bpg/. Accessed: 2023-02-14.

[6] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz. *Deep joint source-channel coding for wireless image transmission*. IEEE Transactions on Cognitive Communications and Networking, 5(3):567–579, 2019.

[7] Q. Mao, T. Yang, Y. Zhang, Z. Wang, M. Wang, S. Wang, L. Jin, and S. Ma. *Extreme Image Compression using Fine-tuned VQGANs*. In Proceedings ot the IEEE Data Compression Conference (DCC), pages 203–212, 2024.

[8] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto. *Learned image compression with discretized gaussian mixture likelihoods and attention modules*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7939–7948, 2020.

[9] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. *Variational image compression with a scale hyperprior*. arXiv preprint arXiv:1802.01436, 2018.

[10] J. Liu, H. Sun, and J. Katto. *Learned image compression with mixed transformer-CNN architectures*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14388–14397, 2023.

[11] P. Esser, R. Rombach, and B. Ommer. *Taming transformers for high-resolution image synthesis*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12873–12883, 2021.

[12] K. Yang, S. Wang, J. Dai, X. Qin, K. Niu, and P. Zhang. *SwinJSCC: Taming swin transformer for deep joint source-channel coding*. IEEE Transactions on Cognitive Communications and Networking, 2024.

[13] H. Wu, Y. Shao, E. Ozfatura, K. Mikolajczyk, and D. Gündüz. *Transformer-aided wireless image transmission with channel feedback*. IEEE Transactions on Wireless Communications, 2024.

[14] G. Wallace. *The JPEG still picture compression standard*. IEEE Transactions on Consumer Electronics, 38(1):xviii–xxxiv, 1992. doi:10.1109/30.125072.

[15] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand. *Overview of the high efficiency video coding (HEVC) standard*. IEEE Transactions on Circuits and Systems for Video Technology, 22(12):1649–1668, 2012.

[16] J. Jiang, K. Zhang, and R. Timofte. *Towards flexible blind JPEG artifacts removal*. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4997–5006, 2021.

[17] L. Zhou, C. Cai, Y. Gao, S. Su, and J. Wu. *Variational autoencoder for low bit-rate image compression*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 2617–2620, 2018.

[18] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool. *Generative adversarial networks for extreme learned image compression*. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 221–231, 2019.

[19] F. Wu, S. Li, and Y.-Q. Zhang. *A framework for efficient progressive fine granularity scalable video coding*. IEEE Transactions on Circuits and Systems for Video Technology, 11(3):332–344, 2001. doi:10.1109/76.911159.

[20] C. Jia, Z. Liu, Y. Wang, S. Ma, and W. Gao. *Layered Image Compression Using Scalable Auto-encoder*. In Proceedings of the 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pages 431–436. IEEE, 2019.

[21] Q. Lan, Q. Zeng, P. Popovski, D. Gündüz, and K. Huang. *Progressive feature transmission for split classification at the wireless edge*. IEEE Transactions on Wireless Communications, 22(6):3837–3852, 2022.

[22] S. Borade, B. Nakiboğlu, and L. Zheng. *Unequal error protection: An information-theoretic perspective*. IEEE Transactions on Information Theory, 55(12):5511–5539, 2009.

[23] A. Goldsmith. *Adaptive Modulation and Coding*, page 283–320. Cambridge University Press, 2005.

[24] A. Hojjat, J. Haberer, and O. Landsiedel. *ProgDTD: Progressive learned image compression with double-tail-drop training*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1130–1139, 2023.

[25] B. Flowers and S. Dey. *BRIC: Bottom-Up Residual Vector Quantization for Learned Image Compression*. IEEE Access, 2024.

[26] W. Cheng, Y. Xiao, S. Zhang, and J. Wang. *Adaptive Finite Blocklength for Ultra-Low Latency in Wireless Communications*. IEEE Transactions on Wireless Communications, 21(6):4450–4463, 2022. doi:10.1109/TWC.2021.3130269.

[27] C. E. Shannon. *A mathematical theory of communication*. The Bell system technical journal, 27(3):379–423, 1948.

[28] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool. *Conditional probability models for deep image compression*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4394–4402, 2018.

[29] X. Lu, H. Wang, W. Dong, F. Wu, Z. Zheng, and G. Shi. *Learning a Deep Vector Quantization Network for Image Compression*. IEEE Access, 7:118815–118825, 2019. doi:10.1109/ACCESS.2019.2934731.

[30] A. Santhirasekaram, A. Kori, M. Winkler, A. Rockall, and B. Glocker. *Vector quantisation for robust segmentation*. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 663–672. Springer, 2022.

[31] L. Zhu, F. Wei, Y. Lu, and D. Chen. *Scaling the Codebook Size of VQGAN to 100,000 with a Utilization Rate of 99%*. arXiv preprint arXiv:2406.11837, 2024.

[32] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. *The faiss library*. arXiv preprint arXiv:2401.08281, 2024.

[33] *Kodak Lossless True Color Image Suite*. http://r0k.us/graphics/kodak/, 1993. Accessed: 2024-10-30.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. *Imagenet: A large-scale hierarchical image database*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.

[35] C. Xiao, Y. R. Zheng, and N. C. Beaulieu. *Novel sum-of-sinusoids simulation models for Rayleigh and Rician fading channels*. IEEE Transactions on Wireless Communications, 5(12):3667–3679, 2006.

8 Conclusions and future work

This dissertation has investigated advanced machine learning techniques to address critical challenges in wireless communications. Our work focused on three interrelated areas: robust parameter estimation for ultra-wideband (UWB) systems, interference cancellation with efficient model compression, and adaptive and progressive learned image compression for noisy channel transmission. In the following sections, we summarize our findings, discuss the overall impact of our contributions, and propose concrete directions for future research.

8.1 Conclusions

In this work, we have developed and evaluated novel machine learning approaches that collectively improve the performance, robustness, and efficiency of wireless communication systems under real-world conditions.

8.1.1 Robust Parameter Estimation

In Part I, we designed a deep convolutional neural network (DCNN) for angle-of-arrival (AoA) estimation in UWB systems. Our model outperformed classical algorithms (e.g., MUSIC and PDoA) by effectively learning to extract discriminative features from channel impulse responses. Notably, we integrated data augmentation and transfer learning techniques, which enabled the model to generalize better to unseen environments—a critical requirement for practical deployments where channel conditions are highly dynamic.

8.1.2 Interference Cancellation and Model Compression

Part II introduced two U-Net–based architectures for blind co-channel interference cancellation. One model incorporated an LSTM bottleneck to capture temporal dependencies, while the other adopted a fully convolutional design. By leveraging fast Fourier convolution blocks, depthwise separable convolutions, and quantization, our architectures achieved significant mean squared error (MSE) improvements with reduced computational complexity and memory footprints. These design choices are especially important for edge devices that operate under strict resource constraints.

8.1.3 Adaptive and Progressive Image Compression

In Part III, we proposed an adaptive and progressive learned image compression (LIC) pipeline tailored for wireless channels affected by noise and fading. By integrating a hyperprior-based model with selective feature masking and a VQGAN-based model employing residual vector quantization, our framework supports partial image reconstruction and iterative quality enhancement. This progressive approach reduces transmission latency and maintains acceptable image quality even under low signal-to-noise ratio (SNR) conditions, as demonstrated through experiments on the Kodak dataset using a Rayleigh fading channel model.

8.1.4 Edge Device Deployment

While this dissertation presents models that are significantly more efficient in terms of inference speed, model size, and MAC operations—along with initial steps toward deployment readiness through quantization and pruning—full integration into commercial edge devices remains a future goal. Target platforms include low-power IoT sensors, mobile devices, and embedded systems like Raspberry Pi or NVIDIA Jetson. Further work is needed to optimize for hardware-specific constraints and validate performance in real-world edge environments, moving from deployment-friendly prototypes to production-ready solutions.

8.1.5 Overall Impact

The overall impact of this work is multifaceted. First, our solutions demonstrate that carefully designed data-driven methods can substantially outperform traditional algorithms in challenging wireless environments, thereby improving both reliability and throughput. Second, by emphasizing model compression techniques—such as advanced quantization, iterative refinement, and early exit strategies—we show that it is possible to deploy high-performance neural networks on resource-constrained devices without sacrificing accuracy. Finally, our progressive image transmission

framework not only meets the stringent latency requirements of real-time applications but also provides a robust mechanism to adapt dynamically to fluctuating channel conditions. These advances collectively pave the way for more adaptive and efficient next-generation communication systems.

8.2 Future Work

While our research has yielded promising results, several specific avenues remain for further investigation:

8.2.1 Enhanced Domain Adaptation and Generalization

Despite the improvements obtained via data augmentation and transfer learning, further work is needed to robustify models against domain shifts. Potential research directions include. Adversarial domain adaptation involves developing an adversarial framework that employs a domain discriminator to align feature distributions between the source (training) and target (deployment) environments, which can help the model learn domain-invariant representations [1]. In addition, self-supervised and meta-learning approaches offer promising strategies by investigating methods such as self-supervised pre-training [2] and meta-learning techniques [3] to enable rapid adaptation with minimal labeled data. For instance, contrastive learning can be employed to learn robust features, while meta-learning facilitates fast adaptation when only a few new examples are available.

8.2.2 Advanced Model Compression and Adaptive Inference

Our current work on quantization has shown promise, but additional compression techniques can further reduce latency and power consumption. In some approaches, the network incorporates multi-stage refinement techniques that generate coarse outputs which are then progressively refined, as described in [4]. This iterative process allows for early termination when the output meets a predefined quality threshold. Additionally, models may be designed with intermediate classifiers or exit points that enable the system to dynamically assess when sufficient confidence has been reached, as noted in [5]. Such early exit architectures help reduce unnecessary computations for simpler inputs, thereby lowering overall inference time.

8.2.3 Comprehensive Performance Metrics

To fully characterize system performance in a real-world deployment, future studies should integrate additional evaluation metrics. Soft log-likelihood ratios (LLRs) extend beyond conventional bit error rate (BER) measures by providing insight into the reliability of the signal processing chain and its impact on downstream decoding

performance [6]. Moreover, evaluating joint metrics that capture both reconstruction fidelity—through measures such as PSNR and SSIM—and communication reliability—using indicators like BER and soft LLR—yields a more comprehensive understanding of the overall system performance [7].

These directions are particularly appealing because they build directly on our current emphasis on lightweight, efficient models for edge deployments while focusing on increasing the adaptability and reliability of machine learning techniques in dynamic wireless environments.

8.2.4 Standardized Datasets

A promising direction for future work lies in developing or adopting standardized datasets across all parts of this research, akin to how the Kodak dataset was utilized in Part III. For communication signals or other specialized applications, carefully curated open-source datasets would enable consistent benchmarking, facilitate reproducibility, and drive more rigorous comparisons between classical and deep learning-based methods. By creating and sharing these datasets, the broader community can more systematically evaluate new approaches and accelerate innovation in wireless communications research.

8.3 Final Remarks

In conclusion, this dissertation has shown that advanced machine learning approaches can significantly enhance the performance, robustness, and efficiency of wireless communication systems. By addressing challenges in parameter estimation, interference cancellation, and image compression, our work not only demonstrates technical improvements over traditional methods but also lays a strong foundation for real-world, adaptive wireless networks. The proposed solutions, together with the targeted future work in domain adaptation, model compression, and comprehensive performance evaluation, point toward a new generation of intelligent communication systems that are both highly efficient and robust to environmental variations.

References

[1] L. Chen, H. Chen, Z. Wei, X. Jin, X. Tan, Y. Jin, and E. Chen. *Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7181–7190, 2022.

[2] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao. *A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[3] A. Vettoruzzo, M.-R. Bouguelia, J. Vanschoren, T. Rögnvaldsson, and K. Santosh. *Advances and challenges in meta-learning: A technical review*. IEEE transactions on pattern analysis and machine intelligence, 46(7):4763–4779, 2024.

[4] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi. *Image super-resolution via iterative refinement*. IEEE transactions on pattern analysis and machine intelligence, 45(4):4713–4726, 2022.

[5] S. Laskaridis, A. Kouris, and N. D. Lane. *Adaptive inference through early-exit networks: Design, challenges and directions*. In Proceedings of the 5th International Workshop on Embedded and Mobile Deep Learning, pages 1–6, 2021.

[6] Y. Zhang, Y. Pan, Q. Tian, Y. Yang, X. Zhang, Y. Zhuang, and W. Gao. *Low Complexity Log Likelihood Ratio Estimation Algorithm Based on Neural Network in Decoding*. In 2023 6th International Conference on Electronics Technology (ICET), pages 652–657, 2023. doi:10.1109/ICET58434.2023.10211434.

[7] S. Tong, X. Yu, R. Li, K. Lu, Z. Zhao, and H. Zhang. *Alternate Learning-Based SNR-Adaptive Sparse Semantic Visual Transmission*. IEEE Transactions on Wireless Communications, pages 1–1, 2024. doi:10.1109/TWC.2024.3512652.