

Learning-based Communication System Design – Autoencoder for (Differential) Block Coded Modulation Designs and Path Loss Predictions

Institute of Sensors, Signals and Systems (ISSS)
School of Engineering and Physical Sciences

Heriot-Watt University, United Kingdom



Ankit Gupta

Supervised by: *Prof. Mathini Sellathurai*

A Thesis Submitted for the Degree of Doctor of Philosophy.

March 2022

The copyright in this thesis is owned by the author. Any quotation from the thesis or use of any of the information contained in it must acknowledge this thesis as the source of the quotation or information.

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Ankit Gupta

Abstract

Shannon's channel coding theorem states the existence of long random codes that can make the error probability arbitrarily small. Recently, advanced error-correcting codes such as turbo and low-density parity-check codes have almost reached the theoretical Shannon limit for binary additive white Gaussian noise channels. However, designing optimal high-rate short-block codes with automatic bit-labeling for various wireless networks is still an unsolved problem.

Deep-learning-based autoencoders (AE) have appeared as a potential near-optimal solution for designing wireless communications systems. We take a holistic approach that jointly optimizes all the components of the communication networks by performing data-driven end-to-end learning of the neural network-based transmitter and receiver together. Specifically, to tackle the fading channels, we show that AE frameworks can perform near-optimal block coded-modulation (BCM) and differential BCM (d-BCM) designs in the presence and absence of the channel state information knowledge. Moreover, we focus on AE-based designing of high-rate short block codes with automatic bit-labeling that are capable of outperforming conventional networks with larger margins as the rate R increases. We also investigate the BCM and d-BCM from an information-theoretic perspective.

With the advent of internet-of-things (IoT) networks and the widespread use of small devices, we face the challenge of limited available bandwidth. Therefore, novel techniques need to be utilized, such as full-duplex (FD) mode transmission reception at the base station for the full utilization of the spectrum, and non-orthogonal multiple access (NOMA) at the user-end for serving multiple IoT devices while fulfilling their quality-of-service requirement. Furthermore, the deployment of relay nodes will play a pivotal role in improving network coverage, reliability, and spectral efficiency for the future 5G networks. Thus, we design and develop novel end-to-end-learning-based AE frameworks for BCM and d-BCM in various scenarios such as amplify-and-forward and decode-and-forward relaying networks, FD relaying networks, and multi-user downlink networks. We focus on interpretability and understand the AE-based BCM and d-BCM from an information-theoretic perspective, such as the AE's estimated mutual information, convergence, loss optimization, and training principles. We also determine the distinct prop-

erties of AE-based (differential) coded-modulation designs in higher-dimensional space. Moreover, we also studied the reproducibility of the trained AE framework.

In contrast, large bandwidth and worldwide spectrum availability at mm-wave bands have also shown a great potential for 5G and beyond, but the high path loss (PL) and significant scattering/absorption loss make the signal propagation challenging. Highly accurate PL prediction is fundamental for mm-wave network planning and optimization, whereas existing methods such as slope-intercept models and ray tracing fall short in capturing the large street-by-street variation seen in urban cities. We also exploited the potential benefits of AE framework-based compression capabilities in mm-wave PL prediction. Specifically, we employ extensive 28 GHz measurements from Manhattan Street canyons and model the street clutters via a LiDAR point cloud dataset and 3D-buildings by a mesh-grid building dataset. We aggressively compress 3D-building shape information using convolutional-AE frameworks to reduce overfitting and propose a machine learning (ML)-based PL prediction model for mm-wave propagation.

Acknowledgements

First and foremost, I would like to express gratitude to my supervisor, Prof. Mathini Sellathurai for her unwavering support, guidance, and invaluable feedback throughout my Ph.D. journey. I am obliged to learn research from her, without her professional expertise, it would not have been possible to complete this work. She always supported me with a positive attitude, patience, and understanding through all the rejections and difficult times till the end. She always believed in me even when I gave up on myself and motivated me. I will always be indebted to her for entrusting in me. She helped me carve a definite path when I was lost and confused in my Ph.D. She became my family away from home.

I would also like to acknowledge the financial support of Heriot-Watt University, EPSRC-UKRI, for my entire Ph.D. tenure.

I would like to express my sincere thanks to Prof. Tharmalingam Ratnarajah for allowing me to begin this amazing journey of research and all the insightful ideas and endless motivation. Also, I would like to extend thanks to Dr. Sudip Biswas, Dr. Hebatallah Shoukry, and Dr. Naveen Mysore Balasubramanya for their thoughtful comments and recommendations during my Ph.D.

I also take this opportunity to thank my colleagues at Nokia-Bell Labs, Dr. Reinaldo Valenzuela, Dr. Jinfeng Du, and Dr. Dmitry Chizhik, for giving me the wonderful opportunity to work on industrial problems and grow as a researcher.

I would like to thank my friends, Abhijeet, Saptarishi, Sovan, Tamal, Shubham, Darius, Yannis, Lucas, and Civan for always supporting me and making this journey wonderful. I would also like to thank my friends, Praveen, Atul, Alankar, Anshul, Akash, Kapish, and Mayur for always being there. I would also like to express a special thanks to Anchita for continuously supporting me through thick and thin.

I also want to present a special thanks to my first mentor, Dr. Keshav Singh, without whom I would not have begun my research career and understood its importance. I would like to thank him for always being there and supporting me as my family.

Last but not the least, I would like to thank my parents, my brother, Arpit, my cousins, Nidhi and Shubham, and all my family members, for supporting me throughout my life and motivating me to fulfill my dream. I would like to present an exceptional thanks to my mother for all her sacrifices, without which I would not have been here today.

Research Thesis Submission

Name:	Ankit Gupta		
School:	EPS		
Version: <i>(i.e. First, Resubmission, Final)</i>	Final	Degree Sought:	Ph.D.

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1. The thesis embodies the results of my own work and has been composed by myself
2. Where appropriate, I have made acknowledgement of the work of others
3. The thesis is the correct version for submission and is the same version as any electronic versions submitted*.
4. My thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5. I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6. I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

ONLY for submissions including published works

7. Where the thesis contains published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) these are accompanied by a critical review which accurately describes my contribution to the research and, for multi-author outputs, a signed declaration indicating the contribution of each author (complete)
8. Inclusion of published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) shall not constitute plagiarism.

* Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.

Signature of Candidate:	<i>Ankit Gupta</i>	Date:	30/03/2022
-------------------------	--------------------	-------	------------

Submission

Submitted By <i>(name in capitals)</i> :	ANKIT GUPTA
Signature of Individual Submitting:	<i>Ankit Gupta</i>
Date Submitted:	30/03/2022

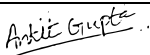
For Completion in the Student Service Centre (SSC)


Limited Access	Requested	Yes	<input type="checkbox"/>	No	<input type="checkbox"/>	Approved	Yes	<input type="checkbox"/>	No	<input type="checkbox"/>
<i>E-thesis Submitted (mandatory for final theses)</i>										
Received in the SSC by <i>(name in capitals)</i> :						Date:				

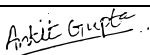
Inclusion of Published Works


Declaration


This thesis contains one or more multi-author published works. In accordance with Regulation 6 (9.1.2) I hereby declare that the contributions of each author to these publications is as follows:


Citation details	A. Gupta and M. Sellathurai, "End-to-End Learning-based Framework for Amplify-and-Forward Relay Networks," <i>IEEE Access</i> , vol. 9, pp. 81660–81677, Jun. 2021.
Author 1	Technical contribution and main author
Author 2	Technical, paper feedback and corrections
Signature:	Ankit Gupta 
Date:	30/03/2022

Citation details	A. Gupta, J. Du, D. Chizhik, R. A. Valenzuela and M. Sellathurai, "Machine Learning-based Urban Canyon Path Loss Prediction using 28 GHz Manhattan Measurements," in <i>IEEE Transactions on Antennas and Propagation, Early Access</i> , Mar. 2022.
Author 1	Technical contribution and main author
Author 2	Technical, paper feedback and corrections
Author 3	Technical, paper feedback and corrections
Author 4	Technical, paper feedback and corrections
Author 5	Technical, paper feedback and corrections
Signature:	Ankit Gupta 
Date:	30/03/2022

Citation details	A. Gupta and M. Sellathurai, "Impact of I/Q Imbalance and Hardware Impairments on Autoencoder-based AF Relay Networks," in <i>IEEE Transactions on Cognitive Communication and Networking, Accepted</i> , Mar. 2022.
Author 1	Technical contribution and main author
Author 2	Technical, paper feedback and corrections
Signature:	Ankit Gupta 
Date:	30/03/2022

Citation details	A. Gupta and M. Sellathurai, "End-to-End Learning-based Amplify-and-Forward Relay Networks using Autoencoders," <i>ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, 2020</i> , pp. 1-6.
Author 1	Technical contribution and main author
Author 2	Technical, paper feedback and corrections
Signature:	Ankit Gupta 
Date:	30/03/2022

Citation details	A. Gupta and M. Sellathurai, "A Stacked-Autoencoder Based End-to-End Learning Framework for Decode-and-Forward Relay Networks," <i>ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020</i> , pp. 5245-5249.
Author 1	Technical contribution and main author
Author 2	Technical, paper feedback and corrections
Signature:	Ankit Gupta 
Date:	30/03/2022

Citation details	A. Gupta and M. Sellathurai, "End-to-End Learning-based Two-Way AF Relay Networks with I/Q Imbalance," in <i>Proc. International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)</i> , pp. 1-5, 2021.
Author 1	Technical contribution and main author
Author 2	Technical, paper feedback and corrections
Signature:	Ankit Gupta 
Date:	30/03/2022

Contents

Contents	viii
List of Tables	xv
List of Figures	xvii
List of Algorithms	xxii
Abbreviations	xxiii
Symbols	xxvii
Notations	xxviii
List of Publications	xxxii
1 Introduction	1
1.1 Summary of Specific Contributions	3
1.2 Thesis Layout	8
2 AE-based Point-to-Point Networks	10
2.1 Introduction	10
2.2 Designing AE Framework for the P2P Networks	14
2.2.1 Designing of NN Encoder	15
2.2.2 Signal Transmission	16
2.2.3 Designing of NN Decoder	16
2.3 AE-based BCM and d-BCM Designs	18
2.4 Training Loss Optimization	19
2.5 Training and Testing Procedure	21
2.5.1 Training Procedure	21
2.5.2 Testing Procedure	22
2.6 Variable AE versus Fixed AE	22
2.6.1 Designing of Variable and Fixed AE Frameworks	22

2.6.2	Numerical Results	26
2.7	Information-Theoretic View of Convergence of the Proposed AE Frameworks	28
2.8	Necessary Conditions for AE Framework’s Convergence	32
2.9	Observations of AE-based BCM and d-BCM Designs	35
2.10	Conclusion	47
Appendices		49
2.A	Autoencoders	49
2.B	Proof of Remark 1	49
3	AE-based Amplify-and-Forward Relay Networks	52
3.1	Introduction	52
3.2	System Model	54
3.3	Proposed AE Frameworks for AF Relay Networks	56
3.3.1	Symbol-Wise AE (SWAE) Framework for AF Relay Networks	56
3.3.2	Bit-Wise AE (BWAE) Framework for AF Relay Networks	57
3.4	Implementation of Autoencoder Framework – Parameters and Training	58
3.4.1	Designing Neural Network-based Encoder	59
3.4.2	Designing AF Relay Node	60
3.4.3	Designing Neural Network-based Decoder	61
3.4.4	Model Training and Updates	62
3.4.5	Predictions	62
3.5	An Information-Theoretic View of Convergence of the Proposed AE Frameworks	63
3.6	Numerical Results	66
3.6.1	AE-based Modulation Design of the AF relay Network under AWGN channels	67
3.6.2	AE-based d-BCM Design under RBF channels	69
3.6.3	AE-based BCM Design under RBF channels	75
3.6.4	Computational Complexity and Time-Cost Analysis	77
3.7	Conclusion	79
4	AE-based Amplify-and-Forward Relay Networks with Hardware Impairments	81
4.1	Introduction	81
4.2	System Model	83
4.2.1	Modelling I/Q Imbalance	83
4.2.2	Signal Transmission Model	84
4.2.3	Without the CSI knowledge	86

4.2.4	Signal-to-interference-and-noise-ratio (SINR) at the Destination node	86
4.3	Proposed Average AE-based Framework for AF Relay Networks with IQI-AHI	87
4.3.1	Designing of the Encoder at the Source Node	91
4.3.2	Designing of the AF Relay Node	91
4.3.3	Designing of the Decoder at the Destination Node	91
4.3.4	Proposed Average AE-based BCM design and d-BCM design Methodologies	92
4.4	Training of the Proposed Average AE-based Framework	93
4.4.1	Neural Network Architecture	93
4.4.2	Training and Testing Dataset Creation	93
4.4.3	Training-Testing Process and Hyper-parameter Settings	94
4.5	Numerical Results	94
4.5.1	Comparison of the Proposed Average AE Framework with Benchmark AE Frameworks	95
4.5.2	Higher Rate Scenario	96
4.5.3	Comparison of the Proposed Average AE Framework with Benchmark AE Frameworks	96
4.5.4	Benchmark Algorithms for Conventional AF Relay Networks with IQI-AHI	98
4.5.5	Average AE-based BCM design with No IQI Compensation	99
4.5.6	Average AE-based BCM design with ZF-based IQI Compensation	100
4.5.7	Average AE-based d-BCM design with No IQI Compensation	101
4.5.8	Average AE-based BCM design and d-BCM design in the presence of only AHI	101
4.5.9	Average AE-based (d-) BCM design with Outer LDPC Codes	102
4.6	Conclusion	104
5	AE-based Two-Way Amplify-and-Forward Relay Networks with Hardware Impairments	106
5.1	Introduction	106
5.2	System Model	108
5.2.1	Signal Transmission Model	108
5.2.2	Impact of IQI on TW-AF relay networks	111
5.2.3	Designing Detector at the Terminal Node	111
5.3	Proposed AE-based TW-AF Relay Network with IQI	112
5.3.1	NN-based Encoder at the Γ^{th} Terminal Node	113

5.3.2	Conventional TW-AF Relay Node	113
5.3.3	NN-based Decoder at the Γ^{th} Terminal Node	114
5.3.4	Process of Creating Training and Testing Datasets	114
5.3.5	Training and Testing Procedure	115
5.3.6	Hyper-Parameter Setting and NN Architectures	116
5.4	Simulation Results	116
5.4.1	Ideal TW-AF Relay Networks with no IQI	116
5.4.2	TW-AF Relay Networks with I/Q Imbalance	123
5.5	Conclusion	126
Appendices		127
5.A	Derivation of Binary CE Loss and Estimated MI for TW-AF Relay networks with IQI	127
6	AE-based Full Duplex Amplify-and-Forward Relay Networks	130
6.1	Introduction	130
6.2	System Model	132
6.2.1	Signal Transmission – Reception Process	132
6.2.2	Modelling Residual Self Interference (RSI)	133
6.2.3	Signal Transmission Model and MLD Decoding	133
6.2.4	Differential FD-AF Relay Networks - Without CSI	134
6.3	Proposed Autoencoder-based FD-AF Relay Networks	135
6.3.1	Designing of NN Encoder at the Source Node	135
6.3.2	Designing of FD-AF Relay Node	136
6.3.3	Designing NN Decoder at the Destination Node	136
6.4	Training of Proposed AE Framework with Hyper-Parameter Setting	136
6.4.1	Training and Testing – Dataset Creation and Methodology	136
6.4.2	NN Architectures	137
6.4.3	Hyper-Parameter Settings	138
6.5	Convergence of the Proposed AE Frameworks for FD-AF Relay Networks	138
6.5.1	Information-Theoretic View of Convergence of Proposed AE Frameworks	139
6.5.2	Necessary Conditions for AE Framework’s Convergence	141
6.6	Analyzing the Observations of AE-based Block Coded Modulation	143
6.7	Numerical Results	149
6.7.1	AE-based d-BCM Design	149
6.7.2	AE-based BCM Design – Perfect CSI Knowledge	151
6.7.3	AE-based BCM Design - Imperfect CSI Knowledge	151
6.7.4	Reproducibility of Proposed AE and d-AE Frameworks	154

6.8	Conclusion	155
Appendices		156
6.A	Conventional FD-AF Relay versus NN-based FD-AF Relay	156
6.B	Impact of Including RTN in AE Frameworks	156
7	AE-based Decode-and-Forward Relay Networks with Hardware Impairments	158
7.1	Introduction	158
7.2	System Model	161
7.2.1	Signal Transmission–Reception between any Two Nodes	162
7.2.2	One-Way Decode-and-Forward Relay Networks with IQI	164
7.2.3	Two-Way Decode-and-Forward Relay Networks with IQI	164
7.2.4	Impact of IQI on OW-DF and TW-DF Relay Networks	166
7.3	Proposed Stacked Bit-wise AE and Denoising AE-based OW-DF Relay Networks with IQI	167
7.3.1	Designing of the Bit-wise AE for Phase 1	169
7.3.2	Designing of the Bit-wise Denoising AE for Phase 2	170
7.3.3	Proposed AE-based BCM and d-BCM Designs	171
7.3.4	Loss Optimization	171
7.3.5	Training Procedure	172
7.3.6	NN Architecture	172
7.3.7	Training and Testing Dataset Creation	172
7.3.8	Hyper-Parameter Settings	173
7.3.9	Testing or Prediction Phase	173
7.4	Proposed Stacked P2P AE-based TW-DF Relay Networks with IQI	174
7.4.1	Designing of Bit-wise P2P AE Framework	175
7.4.2	Loss Optimization and Training Procedure	176
7.4.3	Testing Phase	176
7.5	Performance Evaluation	178
7.5.1	Comparison of Proposed Stacked AE-based frameworks with State-of-The-Art AE works	179
7.5.2	BER Evaluation for the BCM and d-BCM Designs for the OW-DF Relay Networks with IQI	181
7.5.3	BER Evaluation for the BCM and d-BCM Designs for the TW-DF Relay Networks with IQI	186
7.6	Conclusion	187
8	AE-based Multi-User Downlink Networks	188
8.1	Introduction	188

8.2	Conventional Power-Domain Downlink NOMA	192
8.3	Proposed AE-based Multi-user Downlink Network	194
8.3.1	Single Encoder Multiple Decoders (SEMD)	194
8.3.2	Multiple Encoders Multiple Decoders (MEMD)	196
8.3.3	Loss Optimization with QoS Constraint	197
8.3.4	Training and Testing Procedure	198
8.3.5	AE-based (SEMD and MEMD Frameworks) Modulation, BCM and d-BCM Designs	200
8.3.6	NN Architectures of the AE-based SEMD and MEMD Frameworks	200
8.3.7	Hyper-Parameter Settings	201
8.3.8	Advantages of the proposed AE framework over NOMA	202
8.4	Performance Evaluation	203
8.4.1	AE-based Modulation Design	203
8.4.2	AE-based BCM and d-BCM Designs	212
8.4.3	Higher transmission rates	221
8.5	Conclusion	224
9	AE-based Feature Compression and Machine Learning (ML)-based Mm- Wave PL Prediction	226
9.1	Introduction	226
9.1.1	Previous Works	227
9.1.2	Our Contributions	229
9.2	PL Data Collection and Feature Sets Preprocessing	231
9.2.1	28 GHz PL Measurement in Manhattan	231
9.2.2	Street Clutter Modeling using LiDAR Point Cloud Dataset	232
9.2.3	3D-Building Mesh Grid	233
9.3	Street Clutter Feature Extraction from Point Cloud	233
9.3.1	Street Clutter Feature Compression	235
9.3.2	Point Cloud based Expert Street Clutter Features (Clutter)	235
9.4	Autoencoder based Feature Extraction from Building Dataset	237
9.5	ML-based Models and Training-Testing Methodology	239
9.5.1	ML-based models for PL prediction	239
9.5.2	Feature vectors	240
9.5.3	Training and Testing Methodology	241
9.6	Performance Evaluation and Analysis	242
9.6.1	PL Prediction Accuracy (RMSE)	242
9.6.2	Robustness Against Street-by-Street Variation and Distance	245
9.6.3	Feature Importance for the Point cloud-based Street Clutter Features	246

9.6.4	Computational Complexity	248
9.7	Conclusions	248
Appendices		250
9.A	CNN-based Autoencoder design to Compress Building Features	250
9.A.1	Designing the Encoder	250
9.A.2	Designing the Decoder	252
9.A.3	Designing the Loss Function	252
9.B	Loss Convergence and Reproducibility of the Proposed CNN-based AE .	253
9.C	Analyzing the importance of designed Clutter in PL Predictions	254
10	Conclusions and Future Directions	257
10.1	Summary	257
10.1.1	Summary of Various Proposed AE frameworks	257
10.1.2	Special Notes for Channel AE frameworks	260
10.1.3	Special Notes Vision AE frameworks	266
10.2	Future Works	266
Bibliography		267

List of Tables

2.1	Different kinds of AEs with varying rates R considered in the state-of-the-art literature.	12
2.2	NN architecture of <i>Variable AE</i>	23
2.3	NN architecture of <i>Fixed AE</i>	23
2.4	NN architecture of <i>RTN</i>	24
2.5	Training hyper-parameters setting.	24
2.6	Number of optimized parameters and time-cost analysis. Training and testing time is shown in seconds. Testing time is shown for a total of 10^5 blocks of (n, k) data. The number of optimized parameters in RTN is 21, 198.	27
3.4.1	NN architecture of encoder.	59
3.4.2	Process of signal transmission–reception of a block of data (block length n) for conventional AF relay network, and AE-based scenarios – with CSI knowledge and without CSI knowledge using conventional AF relay. . . .	61
3.4.3	Process of signal transmission–reception of a block of data (block length n) for AE-based scenarios without CSI knowledge using power normalized AF relay.	61
3.4.4	NN architecture of decoder.	62
3.6.1	Insights using other metrics for d-BCM design	71
3.6.2	Insights for BCM design with CSI knowledge.	77
3.6.3	Total parameters in the NN-based encoder, decoder and RTN.	77
3.6.4	Time-cost analysis.	78
4.4.1	NN architectures at different nodes in the proposed Average AE framework.	93
4.5.1	Performance evaluation <i>Cases A–E, I</i>	95
5.4.1	Insights for AE-based BCM design for the TW-AF relay networks.	120
6.2.1	Process of signal transmission and reception of $i^{\text{th}} \in \mathcal{U}$ block of data. . . .	133
6.4.1	NN architectures at different nodes in the proposed AE framework.	138

7.1.1 Comparison of proposed Chapter versus state-of-the-art AE literature of DF relay networks	161
7.3.1 NN architectures at different nodes for any of the proposed AE frameworks.	172
8.1.1 Comparison of our chapter with state-of-the-art works	192
8.3.1 NN Architectures for AE-based Modulation designs.	200
8.3.2 NN Architectures for AE-based BCM and d-BCM designs.	201
8.4.1 Minimum squared Euclidean distance of each user $(n, k_j) = (1, 1)$	206
8.4.2 Performance gains of the proposed AE-based SEMD and MEMD frameworks performing modulation designs over the PD-NOMA with conventional and rotated modulation designs, for $E_b/N_0 = 10$ dB at the farthest user.	210
8.4.3 Performance gains of the proposed AE-based SEMD and MEMD frameworks performing BCM designs over the PD-NOMA with conventional and rotated modulation designs with $(7, 4)$ Hamming code, for $E_b/N_0 = 5$ dB at the farthest user.	215
8.4.4 Bit-mapping table for the codewords designed using AE-based SEMD framework.	219
8.4.5 Bit mapping table for the codewords designed using the AE-based MEMD Framework, where X represents the <i>Bit-1</i> (for the d-BCM design) and <i>Bit-3</i> (for the BCM design).	220
8.4.6 Performance gains of the proposed over the conventional models for $n = 7$ at received SNR of 10 dB at the far user for higher transmission rates.	224
9.1.1 Comparison of learning based path loss prediction.	228
9.3.1 Point Cloud based Clutter features $(3, 5, 6, 7)$ for the Manhattan measurement campaign.	236
9.5.1 RMSE in PL prediction.	243
9.6.1 Training and testing complexity of PL prediction algorithms.	247
9.A.1 RMSE in PL prediction using the 4 most important Clutter features ($Clutter_4$).	256

List of Figures

2.1	An illustration of the DL-based AE frameworks for P2P networks.	15
2.2	Illustration of the training of the proposed AE framework.	21
2.3	Performance evaluation for Fixed AE and Variable AE for varying rates (R). In Fig. (a), (c), (e) we analyze BCM designs and in Fig. (b), (d), (f) we analyze d-BCM designs.	25
2.4	Estimated MI for BCM and d-BCM designs using the Fixed and Variable AE frameworks for varying rates R	31
2.5	Proof of the C_1 (second part) and C_2 in Remarks 4 and 5 using the Fixed AE framework for varying rates R	34
2.6	t-SNE representation of the codewords formed in the BCM and d-BCM designs using the Fixed and Variable AE frameworks.	36
2.7	Analyzing the minimum Euclidean distance $d_{E_{\min}}$ for varying (n, k) in a BCM design.	39
2.8	Analyzing the minimum Euclidean distance $d_{E_{\min}}$ for varying (n, k) in a d-BCM design.	40
2.9	Analyzing the BER for varying (n, k) in a BCM design.	40
2.10	Analyzing the BER for varying (n, k) in a d-BCM design.	41
2.11	Minimum Euclidean distance d_E between each a^{th} codeword and its closest v^{th} codeword in a BCM design.	42
2.12	Minimum Euclidean distance d_E between each a^{th} codeword and its closest v^{th} codeword in a d-BCM design.	43
2.13	Normalized second order moment for varying (n, k) in a BCM design.	46
2.14	Normalized second order moment for varying (n, k) in a d-BCM design.	46
2.A.1	An illustration of DL-based Autoencoders.	50
3.2.1	System model for AF relay networks.	54
3.3.1	Symbol-wise autoencoder (SWAE) framework for AF relay networks.	56
3.3.2	Bit-wise autoencoder (BWAE) framework for AF relay networks.	57
3.3.3	Block diagram of training for the proposed bit-wise AE-based end-to-end learning system.	58
3.5.1	Estimated MI versus transmit SNR E_b/N_0	66

3.6.1 Performance evaluation for the AE-based modulation design, with $(n, k) = (1, 2)$, for the AF relay networks.	68
3.6.2 t-SNE representation in 2 dimensions.	69
3.6.3 Conventional versus Proposed AE Frameworks.	71
3.6.4 Bit-wise AE with power normalization (PN) at relay node.	72
3.6.5 Radio transformer network (RTN)	73
3.6.6 RTN network in decoders and comparison with state-of-the-art.	74
3.6.7 Convergence of BWAE in presence of RTN in decoders.	74
3.6.8 AF relay networks with CSI knowledge.	76
4.2.1 AF relay network with I/Q imbalance and aggregate hardware impairments.	83
4.2.2 Impact of IQI and/or AHI on the maximum SINR SINR^{\max} (in dB).	88
4.3.1 Proposed Average AE framework-based AF relay networks with IQI-AHI. Note the <i>Example 2</i> 's 5 th and 6 th bits are shown in a green box.	90
4.5.1 Comparison of the Average AE Framework with Benchmark AE Frameworks.	96
4.5.2 Comparison of the proposed Average AE Framework with benchmark AE frameworks	97
4.5.3 Average AE-based BCM design with no IQI compensation (and with CSI knowledge).	100
4.5.4 Average AE-based BCM design with ZF-based IQI compensation (and with CSI knowledge).	101
4.5.5 Average AE-based d-BCM design with no IQI compensation (and without CSI knowledge).	102
4.5.6 Average AE-based BCM design and d-BCM design in presence of only AHI.	103
4.5.7 Comparison of Average AE-based (d-) BCM design and MLD with (7, 4) Hamming code by using LDPC as outer codes (in Case C).	104
5.1.1 An illustration of OW-AF and TW-AF relay networks implementing the signal transmission–reception between terminal nodes A and B.	107
5.2.1 TW-AF Relay Network with Tx and Rx side I/Q Imbalance at the terminal nodes $\{\Gamma, \Upsilon\} = \{A, B\}$, where $\Gamma \neq \Upsilon$	109
5.2.2 SIR (in dB) versus phase and amplitude offset for Γ^{th} terminal node with $(\xi_T = \xi_R = \xi, \phi_T = \phi_R = \phi)$	111
5.3.1 Proposed AE-based TW-AF relay network with IQI at both the terminal nodes Tx and Rx sides.	113
5.4.1 Comparison of Proposed BWAE v/s state-of-the-art.	117
5.4.2 Estimated Mutual Information versus E_b/N_0 [dB].	118

5.4.3 Comparison of proposed AE framework with different Lambda layers (Cases A–C) in the NN decoder of the terminal nodes.	119
5.4.4 Proposed BWAE v/s Conventional v/s state-of-the-art.	121
5.4.5 Optimal TW-AF relay location for proposed AE framework – Small-scale fading + Path-loss (No shadow fading).	122
5.4.6 Optimal TW-AF relay location for proposed AE framework – Small-scale fading + Path-loss + Shadow fading.	123
5.4.7 BER performance for TW-AF relay networks with I/Q-Imbalance.	124
6.1.1 System model for full-duplex amplify-and-forward relay networks.	132
6.2.1 Proposed AE-based FD-AF relay networks.	135
6.2.2 Signal transmission-reception of i^{th} block of data for the AE-based FD- AF relay network.	135
6.4.1 Block diagram of training for the proposed AE-based end-to-end learning system.	137
6.5.1 Estimated mutual information for proposed AE-based BCM and d-BCM designs, indicating the convergence of the AE frameworks for varying RSI and SNR levels.	140
6.5.2 Proof of the C_1 (second part) and C_2 that AE converges once it designs 2^k codewords.	142
6.6.1 Minimum Euclidean distance $d_{E_{\min}}$ for varying (n, k)	145
6.6.2 BER versus RSI for varying (n, k)	145
6.6.3 Minimum Euclidean distance d_E between each a^{th} codeword and its clos- est v^{th} codeword.	146
6.6.4 Packing density.	147
6.6.5 Average Hamming distance.	148
6.7.1 Performance evaluation for AE-based d-BCM for FD-AF relay networks.	150
6.7.2 Performance evaluation for AE-based BCM for FD-AF relay networks.	152
6.7.3 Impact of the CEQ (ς) on FD-AF relay networks for rate $R = 8/7$	153
6.7.4 Reproducibility of AE-based BCM results for $R = 16/7$ and AE-based d-BCM results for $R = 12/7$	154
6.A.1 Comparing proposed conventional v/s NN-based FD-AF relay-based AE frameworks.	157
6.B.1 Impact of including an RTN in the AE-based BCM design and not includ- ing RTN in d-BCM design.	157
7.1.1 An illustration of OW-DF and TW-DF relay networks implementing the signal transmission–reception between terminal nodes A and B in two and three phase transmissions.	159

7.2.1 Signal transmission between the Tx node (Γ) and the Rx node (Υ) with IQI.	162
7.2.2 One-way decode-and-forward relay networks with IQI.	164
7.2.3 Two-way decode-and-forward relay networks with IQI.	165
7.2.4 Impact of IQI on the SIR (in dB) in OW-DF and TW-DF relay networks.	166
7.3.1 Training of proposed stacked bit-wise AE and bit-wise denoising AE frameworks for OW-DF relay networks with IQI.	167
7.3.2 Testing of proposed stacked bit-wise AE and bit-wise denoising AE frameworks for OW-DF relay networks with IQI.	169
7.3.3 Separate AE frameworks for each transmission phase in the TW-DF relay networks.	173
7.4.1 Training of proposed bit-wise P2P AE framework with IQI.	175
7.4.2 Testing of proposed stacked bit-wise P2P AE framework for TW-DF relay networks with IQI.	177
7.5.1 Comparison of proposed stacked AE frameworks with state-of-the-art AE frameworks in an ideal scenario.	179
7.5.2 Stacked bit-wise AE and denoising bit-wise AE-based BCM designs for OW-DF relay networks with IQI.	180
7.5.3 Stacked bit-wise AE and denoising bit-wise AE-based d-BCM designs for OW-DF relay networks with IQI.	182
7.5.4 t-SNE plots for NN encoder at the relay node in bit-wise denoising AE framework of OW-DF relay network.	183
7.5.5 Stacked bit-wise P2P AE-based BCM designs for the TW-DF relay networks with IQI.	184
7.5.6 Stacked bit-wise P2P AE-based d-BCM designs for the TW-DF relay networks with IQI.	185
8.1.1 An illustration of signal transmission–reception in non-orthogonal multiple access and orthogonal multiple access networks.	189
8.2.1 Conventional power-domain downlink NOMA for multi-user scenario	193
8.3.1 Proposed AE-based single encoder multiple decoder (SEMD) framework.	195
8.3.2 Proposed AE-based multiple encoder multiple decoder (MEMD) framework.	197
8.4.1 Constellation mapping for conventional and rotated NOMA $(n, k_j) = (1, 1)$	204
8.4.2 Constellation mapping obtained for proposed SEMD & MEMD models $(n, k_j) = (1, 1)$	205
8.4.3 Performance evaluation for AE-based SEMD framework performing modulation designs with $(n, k_j) = (1, 1)$	208

8.4.4 Performance evaluation for AE-based MEMD framework performing modulation designs with $(n, k_j) = (1, 1)$	209
8.4.5 BER of each downlink user for $E_b/N_0 = 10$ dB at the farthest user for $(n, k_j) = (1, 1)$ (modulation designs).	211
8.4.6 Performance evaluation for AE-based SEMD framework performing d-BCM design with $(n, k_j) = (7, 4)$	213
8.4.7 Performance evaluation for AE-based MEMD framework performing d-BCM design with $(n, k_j) = (7, 4)$	214
8.4.8 Performance evaluation for AE-based SEMD framework performing BCM design with $(n, k_j) = (7, 4)$	216
8.4.9 Performance evaluation for AE-based MEMD framework performing BCM design with $(n, k_j) = (7, 4)$	217
8.4.10 Bit mapping for AE-based SEMD and MEMD frameworks performing BCM and d-BCM designs with $(n, k_j) = (7, 4)$	218
8.4.11 Autoencoder-based end-to-end learning for higher rates, $n = 7$	222
8.4.12 Autoencoder-based end-to-end learning for higher modulation order, $n = 7$	223
8.4.13 16 QAM + BPSK	224
9.1.1 Methodology adopted in this work.	230
9.2.1 Manhattan measurement data (different color per street).	231
9.2.2 Two streets covered from the same roof top with diverse street clutter density.	233
9.2.3 Aerial-view of W 11 th Street.	234
9.2.4 Example of 3D building mesh grid data	234
9.3.1 Illustration of point cloud based Clutter features 1, 2, 5, 7.	235
9.3.2 Proposed CNN-based autoencoder for building feature compression.	237
9.4.1 Neural network architectures for the proposed CNN-based autoencoder.	238
9.6.1 Street-by-street variation of the average RMSE over 25 runs using <i>Clutter</i> + <i>Building</i>	244
9.6.2 Street-by-street variation.	245
9.6.3 Mean RMSE versus distance.	246
9.6.4 Average feature importance for <i>Clutter</i> feature set using Lasso regression in street-by-street testing, where the error bars indicate the minimum-maximum range over 13 streets.	247
9.A.1 Convergence of CNN-based AE for 13 testing streets.	251
9.B.1 Evaluation of grouped CNNs and iterations needed for reproducibility.	253
9.C.1 Importance of each feature in Elastic-net-based PL prediction model.	254
10.1.1 Summary of various kinds of AE frameworks proposed in this work.	258

List of Algorithms

1	Links-shuffle-split training and testing procedure	241
2	Street-by-street training and testing procedure	241

Abbreviations

0D	Zero dimensional
1D	One dimensional
2D	Two dimensional
3D	Three dimensional
3GPP	3rd Generation Partnership Project
5G	Fifth generation of mobile networks
6G	Sixth generation of mobile networks
AE	Autoencoder
AF	Amplify-and-forward
AHI	Additional hardware impairments
ANN	Artificial neural network
ASR	Achievable-sum-rate
AWGN	Additive white Gaussian noise
BCM	Block-coded modulation
BER	Bit-error-rate
BG2	Base graph 2
BN	Batch normalization
BS	Base station
Build.	Building
BWAE	Bit-Wise AE
CD-NOMA	Code domain NOMA
CE	Cross-entropy
CEQ	Channel estimation quality
CN	Conventional network

CNN	Convolutional neural network
Conv 1D	One dimensional-convolutional layer
Conv-Net-1	Convolutional network 1
Conv-Net-2	Convolutional network 2
CS	Channel statistics
CSI	Channel state information
d	Differential
D	Destination node
d-BCM	Differential block coded modulation
DF	Decode-and-forward
DL	Deep learning
DNF	Denoise-and-forward
DXF	Decode and XOR forward
FD	Full-duplex
FD-AF	Full-duplex amplify-and-forward
GMI	Generalized mutual information
HD	Half-duplex
HDD	Hard decision decoding
HD-AF	Half-duplex amplify-and-forward
I	In-phase
ICT	Information and communication technology
i.i.d.	Independent and identically distributed
Info	Information
IoT	Internet-of-things
IQI	In-phase and quadrature-phase imbalance
IQIC	I/Q imbalance compensation
IQI-AHI	Combined multiplicative I/Q imbalance and additive additional hardware impairments
KL	Kullback-Leibler
LDPC	Low-density parity-check code
LLR	Log-likelihood ratio

LMMSE	Linear minimum mean squared error
LO	Local oscillator
LOS	Line-of-sight
Max-Pooling 1D	One dimensional maximum-pooling layer
MEMD	Multiple encoders multiple decoders
MI	Mutual information
ML	Machine learning
MLD	Maximum likelihood detector
Mm-wave	Millimetre-wave
MSED	Minimum squared Euclidean distance
NLOS	Non-line-of-sight
NN	Neural network
NOMA	Non-orthogonal multiple access
OMA	Orthogonal multiple access
OW	One-way
OW-AF	One-way amplify-and-forward
OW-DF	One-way decode-and-forward
P2P	Point-to-point
PAF	Power allocation factor
PD-NOMA	Power domain non-orthogonal multiple access
PL	Path loss
PN	Power normalization
PNC	Physical network coding
R	Relay node
RBF	Rayleigh block fading
ReLU	Rectifier linear unit
RF	Random forest
RMSE	Root mean squared error
RSI	Residual self-interference
RTN	Radio transformer network

Rx	Receiver
S	Source node
SE	Spectral efficiency
SEMD	Single encoder multiple decoders
SER	Symbol-error-rate
SGD	Stochastic gradient descent
SI	Self-interference
SIC	Self-interference cancellation and Successive interference cancellation
SINR	Signal-to-interference-and-noise-ratio
SIR	Signal-to-interference-ratio
SNR	Signal-to-noise-ratio
SVR	Support vector regression
SWAE	Symbol-Wise autoencoder
Tanh	Tan hyperbolic
TCM	Trellis-coded modulation
TP	Transmit power
t-SNE	t-stochastic neighbour embedding
TW	Two-way
TW-AF	Two-way amplify-and-forward
TW-DF	Two-way decode-and-forward
Tx	Transmitter
UE	User Equipment
Up-Sampling 1D	One dimensional-upsampling layer
w/	With
w/o	Without
WSN	Wireless sensor network
Q	Quadrature-phase
QoS	Quality-of-service
ZF	Zero-forcing

Symbols

$\mathbf{1}_s$	One-hot vector representation of k input bits.
α	Signal amplification factor.
$\sigma_l^{(\cdot)}$	Activation function at the l^{th} dense layer of the (\cdot) node.
$\sigma_{(\cdot)}$	Standard deviation of AWGN at the (\cdot) node.
σ_{ab}	Standard deviation of the channel between the a^{th} and b^{th} node.
$\theta_{(\cdot)}$	NN optimization parameters representing both the weight and bias terms of all the layers at the (\cdot) node.
$\theta_{(\cdot)Tx/Rx}$	NN optimization parameters at the transmitter (Tx) or receiver (Rx) side of the (\cdot) node.
Θ	Total NN optimization parameters in the system model.
$\mathcal{CN}(\mu, \sigma^2)$	Complex normal variable of mean μ and variance σ^2 .
δ_l	Number of neurons in the l^{th} dense layer.
∇	Gradient operator.
τ	Learning rate.
ω_l	Output of l^{th} dense layer.
ω_L	Angular frequency of the local oscillator.
χ	Normalized fourth-order moment or Kurtosis.
$\xi_{(\cdot)}$	Total effective amplitude imbalance at the transmitter $((\cdot) = T)$ or receiver $((\cdot) = R)$ side.
$\phi_{(\cdot)}$	Total effective phase imbalance at the transmitter $((\cdot) = T)$ or receiver $((\cdot) = R)$ side.
μ	Mean.
Γ	Represents the terminal node A or B.
Υ	Represents the terminal node B or A.
$\gamma_{(\cdot)}$	Transmit SNR of the (\cdot) node.

Notations

A	Terminal node.
$\mathcal{A}_{\mathcal{D}}$	Addition layer.
$\mathbf{b}_l^{(\cdot)}$	Bias vector corresponding to the l^{th} dense layer of the (\cdot) node.
B^s	Batch size.
B	Terminal node.
$\mathcal{B}_{\mathcal{N}}$	Batch normalization layer.
CFM	Constellation figure of merit (see Definition 20).
\mathcal{C}	Denotes all the possible alphabets.
$\mathcal{C}_{\mathcal{C}}$	Concatenate layer.
d	Destination node.
$D_{\text{KL}}(p_a \tilde{p}_b)$	Kullback-Leibler (KL)-divergence loss between the true distribution p_a at the NN encoder and the learnt distribution \tilde{p}_b at the NN decoder.
\mathbf{d}_E	Set of minimum Euclidean distances between each a^{th} codeword and its closest b^{th} codeword.
$d_{E_{\min}}$	Minimum Euclidean distance between any of the possible 2^k codewords.
d_E^a	Minimum Euclidean distance between each a^{th} codeword and its closest v^{th} codeword.
E	Number of training epochs.
E_b	Energy per bit.
E_n	Normalized second-order moment.
f	Mapping function
f_L	Frequency of the local oscillator.

$f_{\theta_{(\cdot)}}(\mathbf{a}_{(\cdot)}, \mathbf{b}_{(\cdot)})$	NN-based mapping function at the (\cdot) node (see Definition 6).
$\mathcal{F}_{(\cdot)}$	Input feature vector.
g	De-mapping function.
g_m	Modulation process.
$g_{\theta_b}(\mathbf{y}_b, \tilde{p}_{b\theta_b}(\mathbf{u}_a \mathbf{y}_b))$	NN-based de-mapping function at the b^{th} node (see Definition 9).
(G_1, G_2)	I/Q imbalance parameters at the transmitter side.
$H(\mathbf{u}_{(\cdot)})$	Entropy of the input bits at the (\cdot) node.
h_{ab}	Channel between a^{th} and b^{th} node with zero mean and unit variance, $h_{ab} \sim \mathcal{CN}(0, 1)$.
$H(p_a, \tilde{p}_b)$	Binary cross-entropy loss between true distribution p_a at NN encoder and learnt distribution \tilde{p}_b at NN decoder.
$I_{a\theta_a}(\mathbf{u}_a; \mathbf{y}_b)$	Mutual information between input bits \mathbf{u}_a and received signal \mathbf{y}_b with parameters θ_a .
\mathcal{I}	Estimated mutual information.
$\mathcal{J}(\theta_a, \theta_b)$	Binary cross-entropy loss with NN parameters (θ_a, θ_b) .
k	Number of input-output bits.
\mathcal{K}	Set of different input bits (k).
(K_1, K_2)	I/Q imbalance parameters at the receiver side.
L	Total number of dense layers.
\mathbf{L}_L	Lambda layers.
$l_{(\cdot)}$	Log-likelihood ratio of length m of the \cdot node.
$\mathcal{L}(\mathbf{u}_a, \tilde{p}_{b\theta_b})$	Binary cross-entropy loss between the inputs \mathbf{u}_a and soft-probabilistic outputs \tilde{p}_b (see definition 13).
$\mathbf{LLR}_{(\cdot)}$	Log-likelihood ratio of length m at (\cdot) node.
n	Block length or Number of channel-reuse.
(n, k)	Block of n complex-baseband symbols mapping k bits.
$n_{(\cdot)}$	AWGN at the (\cdot) node with zero mean and $\sigma_{(\cdot)}^2$ variance.
\mathcal{N}	Set of different block lengths (n).
N_0	Noise spectral density.
p_a	True distribution at the NN encoder of the a^{th} node.

$\tilde{p}_{b\theta_b}$	Soft-probabilistic outputs of length k obtained from the NN-based decoder (see Definition 8). Also, referring to the learnt distribution, with parameters θ_b .
\mathbf{P}_N	Power normalization layer (see Definition 7).
$P_{(\cdot)}$	Transmission power of the (\cdot) node.
Q	Denotes the number of samples in a batch of data.
R	Rate or transmission rate.
s	Source node.
S_{train}	Total number of training samples.
S_{test}	Total number of testing samples.
\mathcal{S}	Set of SNR points where we train the AE framework.
\mathcal{S}_T	80% of randomly shuffled training samples, also known as ‘new’ training set.
\mathcal{S}_V	20% of randomly shuffled training samples, also known as validation set.
SINR	Signal-to-interference-and-noise-ratio.
$\mathbf{u}_{(\cdot)}$	Input bits of length k at the (\cdot) node.
$\hat{\mathbf{u}}_{(\cdot)}$	Output bits of length k at the (\cdot) node.
\mathbf{W}_l^a	Weight matrices between the $(l - 1)^{\text{th}}$ and l^{th} dense layers of the a^{th} node.
w	Wright matrix.
$x_{(\cdot)}$	A complex baseband symbol transmitted by the (\cdot) node.
$x_{(\cdot)}^{IQ}$	A complex baseband symbol transmitted in presence of IQI by the (\cdot) node.
$\mathbf{Y}_{(\cdot)}$	Block of n complex baseband symbols received by the (\cdot) node.
$\mathbf{Y}_{(\cdot)}^{IQ}$	Block of n complex baseband symbols with IQI received by the (\cdot) node.
\mathbf{Z}	Total dataset with S total samples and each sample comprises of k bits.

List of Publications

Works Included in the Thesis

Some of the works in this thesis is published or submitted, as detailed below.

- [J1] A. Gupta and M. Sellathurai, “End-to-End Learning-based Framework for Amplify-and-Forward Relay Networks,” *IEEE Access*, vol. 9, pp. 81660–81677, Jun. 2021.
- [J2] A. Gupta, J. Du, D. Chizhik, R. A. Valenzuela and M. Sellathurai, “Machine Learning-based Urban Canyon Path Loss Prediction using 28 GHz Manhattan Measurements,” in *IEEE Transactions on Antennas and Propagation, Early Access*, Mar. 2022.
- [J3] A. Gupta and M. Sellathurai, “Impact of I/Q Imbalance and Hardware Impairments on Autoencoder-based AF Relay Networks,” *IEEE Transactions on Cognitive Communication and Networking, Accepted*, Mar. 2022.
- [J4] A. Gupta, M. Sellathurai and T. Ratnarajah, “End-to-End Learning-based Full-Duplex Amplify-and-Forward Relay Networks using Autoencoders,” *IEEE Transactions on Communications, Under Revision*, Feb. 2022.
- [J5] A. Gupta and M. Sellathurai, “A Stacked Autoencoder-based Decode-and-Forward Relay Networks with I/Q Imbalance,” *IEEE Transactions on Signal Processing, Submitted*, Mar. 2022.
- [J6] A. Gupta and M. Sellathurai, “Revisiting the Autoencoder Frameworks for (Differential) Block Coded Modulation Designs,” *In preparations for Submission*, Mar. 2022.
- [C1] A. Gupta and M. Sellathurai, “End-to-End Learning-based Amplify-and-Forward Relay Networks using Autoencoders,” *ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin*, 2020, pp. 1-6.
- [C2] A. Gupta and M. Sellathurai, “A Stacked-Autoencoder Based End-to-End Learning Framework for Decode-and-Forward Relay Networks,” *ICASSP 2020 - 2020 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 5245-5249.

- [C3] A. Gupta and M. Sellathurai, “End-to-End Learning-based Two-Way AF Relay Networks with I/Q Imbalance,” in Proc. *International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1-5, 2021.
- [C4] René-Jean Essiambre, Roland Ryf, Murali Kodialam, Bin Chen, Mikael Mazur, Juan I. Bonetti, Riccardo Veronese, Hanzi Huang, Ankit Gupta, Faycal Ait Aoudia, Ellsworth C. Burrows, Diego F. Grosz, Luca Palmieri, Mathini Sellathurai, Xi Chen, Nicolas K. Fontaine and Haoshuo Chen, “Increased Reach of Long-Haul Transmission using a Constant-Power 4D Format Designed Using Neural Networks,” *2020 European Conference on Optical Communications (ECOC), Brussels, Belgium, 2020*, pp. 1-4.

Patent Included in this Thesis

We submitted a patent application for approval with UKVI IPO, with details mentioned below. We included this patent in this thesis.

- *Application number:* GB2019616.8
- *Application title:* Deep learning multi-user communication
- *Lodged Date:* 11 December 2020
- *Status:* Pending
- *Link:* <https://www.ipo.gov.uk/p-ipsu/Case/ApplicationNumber/GB2019616.8>

Preworks for the Thesis

The following preworks were done during the Ph.D. period. These preworks focussed on resource allocation using convex optimization techniques that helped in the understanding of relaying, non-orthogonal multiple access, full-duplex, and energy harvesting networks. However, amongst all the machine learning-based applications in the wireless communication designs, the autoencoder frameworks appear most promising. Thus for the sake of brevity, the following pre-works are not included in this thesis.

- [J7] A. Gupta, K. Singh and M. Sellathurai, “Time-Switching EH Based Joint Relay Selection and Resource Allocation algorithms for Multi-user Multi-carrier AF Relay Networks,” in *IEEE Transactions on Green Communications and Networking (Special Section)*, vol. 3, no. 2, pp. 505-522, June 2019.
- [C5] A. Gupta, N.M.Balasubramanya and M. Sellathurai, “Contextual-Bandit based MIMO Relay Selection Policy with Channel Uncertainty”, *ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020*, pp. 1-6.
- [C6] A. Gupta, S. Biswas, K. Singh, T. Ratnarajah and M. Sellathurai, “An energy-efficient Approach towards power allocation in NOMA Full-duplex AF Relay Systems,” in *Proc. IEEE International Workshop on Signal Processing Advances in Wireless Communications (IEEE SPAWC)*, pp. 1-5, 2018.
- [C7] N.M. Balasubramanya, A. Gupta, and M. Sellathurai, “Combining Code-domain and Power-domain NOMA for Supporting Higher Number of Users,” in *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 1-5, 2018.
- [C8] A. Gupta and M. Sellathurai, “Indoor Positioning based on Deep Convolutional Neural Networks,” Poster presented in *IEEE Communication Theory Workshop (IEEE CTW)*, Iceland, 2019.

Chapter 1

Introduction

Today, the communication system is well-defined into various independent blocks, each of which performs an explicit function, such as source encoding, channel encoding, modulation, demodulation, channel decoding, and source decoding [1]. Designing the modulation schemes and error-correcting codes in a traditional digital communication system are viewed as two separate problems. For instance, the coding theorists focus on improving the performance by designing the error-correcting codes by aggregating the impacts of modulation and channel impairments into an “effective physical channel” having defined analytical characteristics. Likewise, communication theorists focus on band-limited channels by commonly overlooking the error-correcting codes while devising effective modulation schemes to conserve the bandwidth. However, there is no practical evidence that individually optimizing each block of the communication network achieves the best possible throughput. For example, the separate source and channel coding and the separate channel coding and modulation designs are known to be suboptimal [2], [3].

In the seminal 1982 work [4], Ungerboeck introduced the bandwidth-efficient coded modulation design by combining the convolutional coding with modulation for achieving the optimal designs, thereby enhancing the reliability of the network. Broadly, we can classify the coded modulation designs into two categories – trellis-coded modulation (TCM) design and block-coded modulation (BCM) design. In a TCM design, a trellis is employed to specify the dependency of each codeword on any earlier codewords. The dependency is employed to improve the minimum squared Euclidean distance (MSED) between the different code paths comprised of the codewords [5]. In a BCM design, for a block length (n), the codewords are selected in the $2n$ -dimensional space, such that the MSED between any two codewords is sufficiently large to achieve the desired transmission reliability [6]. Moreover, the BCM designs can provide high-performance gains and low complexity design in contrast to the TCM designs [7]. However, even the most famous TCM and BCM designs for the binary Additive white Gaussian noise (AWGN) channels experience deteriorating performance in fading channels [8].

Moreover, with the advent of the ultra-dense internet-of-things (IoT) and wireless sensor network (WSN), reducing the energy requirement has become pivotal, and employment of ultra-short block codes has gained considerable traction in industrial application, especially for small devices.

Designing optimum BCM for short block lengths with high transmission rates remains a challenging problem, which becomes even more complicated under fading channels, such as Rayleigh block fading (RBF) channels. This is because the BCM design is closely tied to the sphere packing problem, which presents three significant challenges. Firstly, the maximum number of codewords increases exponentially with the number of bits (k) as 2^k , and the problem of fitting these 2^k codewords in $2n$ -dimensional space (where n denotes the block length) becomes complicated. Secondly, it is challenging to determine the number of neighbors for each codeword optimally. Lastly, the bit-labeling of the designed codewords in $2n$ -dimensional space remains unsolved. Although, the existence of the optimum random short BCM design in $2n$ -dimensional space is well established [9]. Still, no method exists to obtain these optimum random BCM designs, even for binary AWGN channels. Under a fading channel, such as RBF channels, the BCMs are designed with channel state information (CSI) knowledge. However, the CSI estimation increases the feedback overhead, especially with the advent of IoT networks, the feedback overhead will manifold exponentially. Further, IoT network imposes ultra-reliable and low-latency requirements. Thus, it is also pivotal to design the differential BCM (d-BCM) without the CSI knowledge. This thesis focuses on RBF channels and strives to answer the problem.

“What can be accomplished by performing BCM and d-BCM designs with a focus on the ultra-short block-length (n) regime with higher transmission rates (R)?”

This inquest is also of the latest interest to IoT devices, usually communicating periodic and short messages with low-latency requirements.

Recently, the information and communication technologies (ICT) sector envisioned in [10], [11] that the applications of machine learning (ML) and deep learning (DL) in the physical layer will have a paradigm shift from an enhancement to a foundational block via the self-organizing transmitter and receiver architectures performing end-to-end learning, also known as autoencoder (AE) architectures, as introduced by O’Shea in 2017 in the seminal work [12]. The AE framework for wireless communication networks (also called “Channel AE”) is inspired by the traditional DL-based AE frameworks employed in the Vision community, hereby referred to as “Vision AE.” The difference is that the Vision AE framework aims to compress the input data by removing its redundancy. At the same time, the Channel AE of the communication networks often adds redundancy to the input data, such that the encoded data becomes robust to propagation channel and hardware impairments.

The main advantage of employing the Channel AE frameworks is that we do not

require detailed waveform design, constellation mapping, and/or reference signals. By performing end-to-end training between the input signal at the neural network (NN) of the transmitter and decoded signal from the NN of the receiver, the AE framework can learn to determine the best parameters. Thus, this thesis aims to solve

“How can we perform the BCM and d-BCM designs using the Channel AE frameworks under a fading channel?”

Moreover, the current signal processing algorithms are mathematically tractable, with a solid information theory background. However, a practical system faces multiple distortions because of the non-linear elements and hardware imperfections that a conventional algorithm cannot capture [13]. For example, in-phase (I) and quadrature-phase (Q) imbalance (IQI), power amplifier non-linearities, and phase noise, leading to undesirable effects such as crosstalks, an added image signal, frequency interference, etc. At the same time, the NNs employed in DL algorithms are universal function approximators that are highly parallelized and do not require exact data [14]. Thus, extensive data employed during the training of the AE framework will directly include hardware imperfections that the NNs can learn at the transmitter and receiver. Thus, this thesis also aims to determine

“What will be the role of the Channel AE-based BCM and d-BCM designs in removing the deteriorating impacts of the hardware impairments?”

In general, the DL-based AE frameworks (both Channel and Vision AEs) are considered the “black box”, where the insights to the obtained solutions almost remain non-existent. Several pioneering works [12], [15], [16] have started focusing on opening the black-box of the Channel AEs. Moreover, in the seminal work of 1991 [9], Oliveira investigated random short BCM design in $2n$ -dimensional space using the MSED metric. Nevertheless, the Channel AE-based BCM and d-BCM designs remain a black box. Thus, in this thesis, we also aim to answer

“How can we open the black-box of the Channel AE-based BCM and d-BCM designs?”

Lastly, we focus on the capabilities of the Vision AE framework for reducing the high-dimensional data to a minimal number of compressed features. In summary, we aim to understand the potential benefits of AE frameworks (both Channel and Vision AEs) in the learning-based communication system design.

1.1 Summary of Specific Contributions

This thesis focuses on the learning-based communication system design by employing AE frameworks. We study the traditional Vision AE framework for high-dimensional feature compression utilized in ML-based mm-wave path-loss (PL) predictions in Chapter 9, and the Channel AE framework for high-dimensional BCM and d-BCM designs in

Chapters 2–8, for high transmission rates $R = k/n$ [bits/channel-reuse] and short block lengths (n). Please note for the sake of readability, we refer to the Vision and Channel AEs as simply “AE” hereafter. The most important contributions are summarized below.

We begin by studying the AE-based frameworks in point-to-point (P2P) communication networks in Chapter 2. Firstly, we conduct an in-depth literature survey and analyze the shortcomings of the state-of-the-art literature. Secondly, based on the number of neurons in the dense layers and input-output bits (k) of the AE, we classify the AE frameworks as Fixed AE and Variable AE frameworks. Then, we propose both of these AE frameworks performing BCM and d-BCM designs for high rate transmission and short block lengths. Later, we compare the proposed AE frameworks in terms of the BER performance, optimized parameters, memory usage, and time-complexity to determine the best possible NN architecture designing method for varying transmission rates. Thirdly, we focus on the convergence of the training of both the proposed AE frameworks – (i) we analyze the convergence of the proposed AE frameworks by estimating the mutual information (MI) between the input-output of the AE for varying signal-to-noise-ratio (SNR) levels, and (ii) we provide the necessary conditions for AE’s convergence concerning the training samples and minimized cross-entropy (CE) loss. Lastly, we focus on opening the black-box of the AE-based BCM and d-BCM designs by revealing the distinct properties of codewords designed in higher-dimensional space by the proposed AE frameworks. Specifically, we utilize the minimum Euclidean distance, packing density, and Kurtosis to understand the designed codewords’ properties. Furthermore, we compare the properties of the AE-based BCM and d-BCM designs.

Relay networks have appeared as an innovative technology in the past decade owing to their potential of increasing network coverage, reliability, and capacity [17], [18]. Moreover, amplify-and-forward (AF) relaying is adopted practically because of its low implementation complexity. In Chapter 3, we study the AE-based AF relaying network. Unlike the literature that employs a neural network (NN)-based relay node with full CSI knowledge, we consider a conventional relay node that only amplifies the received signal using CSI gains. We propose and compare the symbol-wise and bit-wise AE frameworks by minimizing categorical and binary CE loss that maximizes the symbol-wise and bit-wise MI, respectively. Furthermore, we determine the estimated MI and examine the convergence of both AE frameworks with SNR. For both these AE frameworks, we design BCM and d-BCM designs that obtains codewords in higher-dimensional space. To explain the properties of the codewords designed by bit-wise and symbol-wise AE frameworks, we utilize various metrics like minimum Euclidean distance, normalized second-order and fourth-order moments, and constellation figures of merit. For the d-BCM designs – (i) we also propose and investigate the impact of an NN-based radio transformer network (RTN) on the training and bit-error-rate (BER) performance of the network, and (ii) we employ

power normalization-based amplification that normalizes the transmission power of each block of symbols to the block length.

In practice, any communication system is compromised by the IQI, deteriorating the network performance [19]. Apart from the IQI that has a multiplicative effect on the signal, tackling the other additional hardware impairments (AHI) (also referred to as residual hardware impairments) that have an additive effect on the signal is of notable importance [13]. In Chapter 4, we propose a novel Average AE-based AF relay network impacted by the multiplicative IQI and additive AHI, where the source and destination nodes are equipped with an NN-based encoder and decoder. Also, we consider a conventional AF relay node that assists the transmission. The average AE employs multiple small NN-based decoders at the destination node, each decoding a soft probabilistic output that is averaged to obtain the final soft probabilistic output at the destination node. By considering multiple small NN decoders, we reduce the implementation complexity significantly while improving the performance compared to the AE with a single large NN-based decoder. Within this Average AE framework, we propose a BCM design with zero-forcing (ZF)-based IQI compensation that considers the CSI and IQI knowledge availability. However, the IQI and CSI need to be estimated separately. Thus, we also propose a BCM design with no IQI compensation that requires only the CSI knowledge. Finally, we propose a d-BCM design that removes the necessity of both the CSI and IQI knowledge. Under low signal-to-interference-and-noise-ratio (SINR) regimes, we show that the proposed Average AE framework outperforms the optimal maximum likelihood detector (MLD) by a considerable margin, even without the IQI parameters information, reducing the feedback overhead.

A two-way (TW) relay network overcomes the drawbacks of the half-duplex one-way (OW) relay network by utilizing the spectrum resources more efficiently [20, 21, 22, 23]. In particular, in a TW relay network, both the source and destination nodes, widely referred to as terminal nodes, communicate with each other in two transmission phases, thereby improving the spectral efficiency (SE) by 100% compared to an OW relay network proposed in Chapters 3, 4. In Chapter 5, we propose AE-based TW-AF relay networks in the presence of IQI at the terminal nodes. We consider an NN-based terminal node and a conventional TW-AF relay node to maintain the minimal implementation cost at the relay node. Specifically, we jointly perform BCM design in higher dimensional space for both the terminal nodes to tackle the interference from simultaneously received signals at the TWAF relay node and the deteriorating impacts of the IQI at the terminal nodes. Further, we determine the best pre-processing technique at the NN decoder to remove the self-interference and decode the signals. We utilize various performance metrics for higher interpretability in the BCM designs, such as minimum Euclidean distance, second- and fourth-order moments, and constellation figures of merit. Lastly, we determine the

optimal location for the TW-AF relay in an AE framework under the presence of path-loss and shadow fading effects.

Recently, the full-duplex (FD) mode is recognized as an enabling technology to realize the expected gains in the future networks, as it can double the SE by establishing concurrent transmission and reception on the same temporal and spectral resources [24]. Although, the superior self-interference cancellation (SIC) methods have attested to the facilitation of FD transmission. However, even with multiple SIC techniques, a residual self-interference (RSI) is always present in the system, deteriorating the FD performance. In Chapter 6, we study AE-based FD-AF relay networks under the presence of RSI for high transmission rates (R). In particular, we design an NN-based encoder and decoder at the source and destination nodes, respectively, with a conventional FD-AF relay node assisting the transmission. Depending on the availability of the CSI knowledge, we propose AE-based BCM or d-BCM. We also investigate the performance of the proposed AE-based BCMs under the imperfect CSI knowledge. Moreover, we analyze the training convergence of the proposed AE frameworks by determining the estimated MI between the input-output of the AE. We show that for any given hyper-parameter settings, the AE converges when the NN encoder at the source node has designed 2^k codewords, and the CE loss is converged. Furthermore, we reveal five distinct properties of the codewords designed by the proposed AE framework in the higher-dimensional space by analyzing Euclidean distance, packing density, Hamming distance, and Kurtosis of the designed codewords. Lastly, we also show that the proposed AE frameworks are highly reproducible even with different training samples and weight initialization.

Although the AF relay provides low complexity re-transmission of the amplified signal, it also amplifies the noise at the relay node [25]. Traditionally, to remove the noise amplification, decode-and-forward (DF) relay is adopted, which has added advantage of using different coding designs at the source and relay nodes. In Chapter 7, we propose a stacked AE-based DF relay network in the presence of IQI at all the nodes. In particular, we propose a stacked AE-based BCM and d-BCM designs for both the one-way DF (OW-DF) and two-way DF (TW-DF) relay networks with IQI. For the OW-DF relay networks, we propose a stacked bit-wise AE and a novel bit-wise denoising AE-based BCM and d-BCM designs. Furthermore, we propose a new training policy for the bit-wise denoising AE framework with a two-step training approach. Moreover, we propose a novel single-step training policy for the TW-DF relay networks. Specifically, we propose to train bit-wise AE-based BCM and d-BCM designs for P2P networks and employ a trained bit-wise P2P AE framework in a novel stacked form to design the TW-DF relay networks. For both the OW-DF and TW-DF relay networks, we design a single stacked AE framework that can generalize well on varying levels of IQI and signal-to-noise ratio (SNR). We also remove the impact of IQI, even without utilizing the IQI parameters

information, reducing the feedback overhead.

Designing a multiple access technique is one of the requisite notions of improving the system capacity. In Chapter 8, we study an AE-based framework for downlink multi-user networks considering the non-orthogonal multiple access (NOMA) networks as the benchmark. First, we utilize an individual decoder for each downlink user, whereas the base station (BS) has single or multiple encoders to superpose the signals. Therefore, based on the number of encoders employed at the BS, we design two architectures - single encoder multiple decoders (SEMD) and multiple encoders multiple decoders (MEMD). Second, we design the AE frameworks to remove the necessity of performing the power allocation factor-based signal superposition at the BS and successive interference cancellation (SIC) at strong users, unlike conventional NOMA. Thirdly, both the AE-based SEMD and MEMD frameworks perform BCM and d-BCM design. As a result, we can decode the received signal at each user without using the CSI knowledge, unlike the conventional NOMA, where CSI remains mandatory for the successful implementation of the SIC. Further, we analyze the bit mapping obtained by the AE to investigate the gains achieved by the proposed AE frameworks. Lastly, we show that our proposed frameworks perform better with the increasing number of multiplexed users and modulation order.

Until now, in this thesis, we focused on designing solutions for limited bandwidth scenarios. However, Large bandwidth and worldwide spectrum availability at cm/mm-wave bands have shown excellent potential for 5G and beyond, but the high path loss (PL) and significant scattering/absorption loss from various street clutter make link budget challenging. Highly accurate and site-specific PL prediction is fundamental for cm/mm-wave network planning and optimization, but existing methods such as empirical slope-intercept models and deterministic ray-tracing fall short in capturing the sizeable street-by-street variation seen in urban cities. Machine learning (ML)-based PL prediction for cm/mm-wave bands is promising but faces three key challenges: 1) reliability due to lack of measurement data; 2) generalizability due to lack of extrapolation; 3) interpretability due to complex ML models. In Chapter 9, we propose an ML-based PL prediction model based on extensive 28 GHz measurements from Manhattan street canyons where street clutters such as foliage, trees, and lampposts are modeled via a LiDAR point cloud dataset, and the buildings are modeled by a mesh-grid building dataset. We extract expert knowledge-driven features from the point cloud to improve interpretability and compress 3D building information using a novel convolutional NN (CNN)-based AE framework to reduce overfitting. Simple linear and non-linear ML methods are trained using these features based on a new street-by-street training and testing procedure to enforce generalizability such that the trained ML methods could predict PL on a never-measured street. We show that the proposed model achieves a prediction error (RMSE) of 4.8 ± 1.0 dB compared to 11.1 ± 4.5 dB and 6.6 ± 2.1 dB for 3GPP LOS and slope-intercept fit, where

the standard deviation indicates the street-by-street variation.

1.2 Thesis Layout

The thesis is organized as follows:

Chapter 1 provides the motivation of research in this thesis as well as a brief overview of the research and structure of the thesis.

In Chapter 2, we study the AE-based BCM and d-BCM designs for a P2P communication network. Firstly, we present a literature review. Then, we propose and compare the Fixed and Variable AE frameworks. Also, we analyze the training convergence of the AE frameworks and the distinct properties of the AE-based BCM and d-BCM designs.

In Chapter 3, we investigate the AE-based BCM and d-BCM designs for an AF relay network. We also propose and compare the bit-wise and symbol-wise AE frameworks. Finally, we also evaluate the benefits of the RTN in a d-BCM design.

In Chapter 4, we consider the AE-based BCM and d-BCM designs for an AF relay network impacted by the hardware impairments (IQI and/or AHI). First, we introduce a novel Average AE framework with multiple small NN decoders. Then, we propose BCM and d-BCM designs to remove the necessity of IQI parameter information.

In Chapter 5, we study the AE-based BCM design for a TW-AF relay network impacted by the hardware impairments (IQI). We analyze the importance of performing complex domain operations in the Lambda layer and the optimal relay location.

In Chapter 6, we investigate the AE-based BCM and d-BCM designs for an FD-AF relay network in the presence of the RSI. We analyze the BCM designs with channel estimation error. We also investigate the reproducibility of the AE frameworks. Finally, we also analyze the training convergence and properties of the AE-based BCM design.

In Chapter 7, we consider the AE-based BCM and d-BCM designs for the OW-DF and TW-DF relay networks impacted by the hardware impairments (multiplicative IQI). First, we propose a Stacked AE framework-based approach. Second, we also propose a bit-wise denoising AE framework and its training policy. Finally, we propose BCM and d-BCM designs to remove the necessity of IQI parameter information.

In Chapter 8, we study the AE-based BCM and d-BCM designs for a multi-user downlink network. First, we propose AE-based SEMD and MEMD frameworks. Then, we consider the conventional NOMA as a benchmark and remove its drawbacks. Lastly, we investigate the bit-labeling of the designed codewords.

In Chapter 9, we design expert features with physical meaning from the LiDAR point cloud dataset. We also propose a novel CNN AE-based building feature compression. Lastly, we propose an ML-based PL prediction model for mm-wave propagation in an urban street canyon using these feature sets and a new training-testing policy.

Finally, in Chapter 10, we provide conclusions of this thesis and suggest possible research extensions and directions.

Chapter 2

AE-based Point-to-Point Networks

2.1 Introduction

From information theory perspective, we can define the rate as follows:

Definition 1 (Rate (R) [26]) *The rate R can be defined as the number of bits transmitted (denoted by k) divided by the number of channel uses (denoted by n), given by*

$$R = k/n \quad [\text{bits/channel reuse}] \quad (2.1)$$

The autoencoder (AE)¹ frameworks have appeared as a potential solution for performing data-driven designs by end-to-end learning of the neural network (NN)-based source and destination nodes together, for any transmission rate (R) [12]. The bit-decoding problem in the wireless communication network is considered a classification problem in the AE framework. Thus, the AE framework is optimized by minimizing the cross-entropy (CE) loss between the input at the NN encoder and output at the NN decoder. Depending on the input-output of the AE, we can broadly classify the AE frameworks as follows:

Definition 2 (Symbol-wise AE [12]) *Herein, the NN encoder's input and the NN decoder's ground-truth output is a 2^k one-hot vector representation of the k bits, wherein only one element is unity at a time, indicating one of the possible symbols. Thus, the NN decoder employs Softmax activation function in its last layer, and the symbol-wise AE framework is optimized by minimizing the categorical CE loss.*

Definition 3 (Bit-wise AE [27]) *Herein, the NN encoder's input and the NN decoder's ground-truth output are k bits. Thus, the NN decoder employs Sigmoid activation function*

¹As detailed in Chapter 1, the Channel AE framework for wireless communication networks is inspired from the traditional deep-learning (DL)-based Vision AE frameworks as detailed in Appendix 2.A. We focus on Channel AEs in this Chapter.

in its last layer, and the bit-wise AE framework is optimized by minimizing the binary CE loss.

As shown in Table 2.1, the symbol-wise AE frameworks have been widely investigated for point-to-point (P2P) communication networks [12, 28, 29, 30, 31, 32, 33, 34], for the multi-user networks [35], and for relay networks [36, 37]. However, as the symbol-wise AE's input and output are in the form of symbols, bit-labeling has to be done separately either by exhaustively searching through the $2^k!$ combinations or by heuristic search method, leading to the sub-optimal bit-labelings and bit-error-rate (BER) performance [27]. Recently the bit-wise AE framework has been proposed for P2P networks [27], and relay networks [38, 39]. Although the bit-wise AEs seem like a trivial modification of the symbol-wise AEs, we obtain automatic bit-labeling by providing the bit-wise AE's input and output in the form of bits. Further, a comparison of bit-wise and symbol-wise AEs-based 2-dimensional (I and Q) modulation design showed that the symbol-wise AE-based trained constellation improves the symbol-error-rate (SER), it degrades the BER in comparison to the bit-wise AE in a P2P network. This is because symbol-wise AE aims to maximize the symbol-wise mutual information (MI). In contrast, bit-wise AE aims to maximize the bit-wise MI [27].

Analyzing the current literature in Table 2.1, we find that the NNs employed in the AE frameworks are usually comprised of the dense layers². Also, two methods have appeared to design the NN architectures of the AE framework employed in the physical layer, which we broadly define as *Fixed AE* and *Variable AE*, for the first time in this chapter, depending on the number of neurons (δ_l) in the l^{th} dense layer of the AE, detailed as follows:

Definition 4 (Variable AE) *In this AE framework, we consider that the number of neurons in each of the l^{th} dense layer (δ_l) in the NN is directly dependent on the number of input-output bits (k) or the rate $R = k/n$. In particular, we keep $\delta_l = 2^k$.*

Definition 5 (Fixed AE) *In this AE framework, we decide a fixed number of neurons in each of the l^{th} dense layer (δ_l) in the NN, such that it can handle various combinations of input-output bits (k) or the rate $R = k/n$. In particular, δ_l remains constant for any rate R .*

Moreover, the AE frameworks are also called end-to-end learning-based frameworks because all the components at the source and destination nodes are optimized together in an end-to-end manner. While, the AE is trained by minimizing the CE loss between the input and output of the AE at the source and destination nodes, respectively.

²Wherein the neurons of any dense layer are fully connected with the neurons of the previous and next dense layer. This results in the formation of the weight matrices utilized for mapping and de-mapping in the NN-based encoder and decoder at the source and destination nodes.

Table 2.1: Different kinds of AEs with varying rates R considered in the state-of-the-art literature.

Ref	Scenario	Symbol-wise AE	Bit-wise AE	Fixed		Variable AE	AE-based Block Modulation Design			AE-based Block Coded Modulation Design							
				AE	AE		AWGN	With CSI	RBF	Without CSI	AWGN	With CSI	RBF	Without CSI			
[12]	P2P and Two-user; Dense	✓	✗	✗	✗	✓	1, 2	–	–	–	–	–	–	–	–	4/8	
[27]	P2P; Dense	✓	✓	✗	✗	✓	2, 3, 4, 6, 8	6, 8	–	–	–	–	–	–	–	–	–
[31]	P2P; Dense	✓	✗	✓	✗	✗	–	–	2	2	–	–	–	–	–	–	–
[34]	P2P; Dense, Conv.	✓	✗	✗	✗	✓	2	2	2	2	–	–	–	–	–	–	–
[15]	P2P; Dense	✗	✓	✓	✓	✗	2, 4	–	–	–	–	–	–	–	–	–	–
[38]	Two-way AF relay; Dense	✓	✗	✓	✓	✗	2	2	–	–	–	–	–	–	–	–	–
[36]	One-way DF relay; Dense	✓	✗	✗	✗	✓	2	–	–	2	–	–	4/7	–	–	–	–
[39]	Cooperative NOMA; Dense	✗	✓	✓	✓	✗	2	2	–	–	–	–	–	–	–	–	–
[32]	P2P; Conv.	✓	✗	✗	✗	✓	1, 2, 3, 4, 6	1, 2, 3, 4, 6	1	1	–	–	4/3	–	–	–	–
[28]	P2P; Dense, Conv	✓	✗	✗	✗	✓	–	–	–	–	–	–	4/7	–	–	–	–
[29]	P2P; Dense	✓	✗	✗	✗	✓	–	–	–	–	–	–	3/7, 4/7, 5/7, 6/7	–	–	–	–
[30]	P2P; Dense	✓	✗	✗	✗	✓	–	–	–	–	–	–	4/7	–	–	–	–
[33]	P2P Variational AE; Dense	✓	✗	✓	✓	✗	2	2	–	–	–	–	–	–	–	–	–
[35]	P2P and Two-user; Dense	✓	✗	✗	✗	✓	1, 2, 3, 4	–	–	–	–	–	–	–	–	–	–
[40]	Multi-carrier; Dense	✓	✗	✗	✗	✓	1, 2.5, 3	1, 2.5, 3	–	–	–	–	–	–	–	–	–

As a result of the end-to-end training in an AE-based communication network, the NN encoder at the source node learns a mapping function (f) to perform joint block channel coding and modulation design together with the NN decoder at the destination node learning a de-mapping function (g) to perform joint block channel decoding and demodulation design, also referred to as *AE-based block coded modulation (BCM) designs* [12]. However, the perfect estimation of channel state information (CSI) knowledge remains a challenging task; even estimation of CSI knowledge increases feedback overhead. Furthermore, with the advent of internet-of-things (IoT) and wireless sensor networks (WSNs) small devices, it becomes important to reduce latency and the feedback overhead for CSI estimation will manifold exponentially. Thus the BCM design using the AE frameworks in the absence of the CSI knowledge have also motivated the *AE-based differential BCM (d-BCM) designs* [12].

As detailed in Table 2.1, until now, only a handful of works have focussed on AE-based (differential) block coded-modulation designs using only a ‘symbol-wise AE’ framework – (i) under additive white Gaussian noise (AWGN) channels: for rates $R = 2/7, 4/7, 6/7$ in [12], [28], [29], (ii) under Rayleigh block fading (RBF) channels: for rate $R = 4/8$ in [12], rate $R = 4/7$ in [36]. Moreover, although the NN-based AE frameworks provide us a methodology to obtain these BCMs, they are still treated as a “black-box”. Furthermore, these works have considered low rate transmission while showing significant BER performance gains of the AE frameworks over the BPSK or differential (d)-BPSK modulation-demodulation and (7, 4) Hamming code as baseline channel-coding technique. Thus, it becomes pivotal to develop AE frameworks for (differential) block coded-modulation designs that cater to the high rate requirements of future IoT and WSNs for short block lengths requiring low memory and implementation complexity.

Later, in *Chapter 3*, we focus on comparing the bit-wise and symbol-wise AE-based BCM and d-BCM design, wherein we show that the bit-wise AE outperforms the symbol-wise AE-based BCM and d-BCM designs. Thus, we consider a bit-wise AE in this chapter, and the major contributions of this chapter are detailed below.

- We propose both the ‘Fixed and Variable bit-wise AE’ frameworks for P2P communication networks under an RBF channel for short block length ($n = 7$), performing:
 - BCM designs for higher rates such as $R = 4/7, 8/7, 16/7$, in comparison to the conventional BPSK, QPSK, and 16-QAM modulations with (7, 4) Hamming code as baseline error correction code.
 - d-BCM designs for higher rates such as $R = 4/7, 8/7, 12/7$, in comparison to the conventional d-BPSK, d-QPSK, and d-PSK-8 modulations with (7, 4) Hamming code as baseline error correction code.

We compare the proposed Fixed AE and Variable AE frameworks in terms of the BER performance, the number of optimized parameters, memory usage, and time-complexity to determine the best possible NN architecture designing method for varying transmission rates.

- Focussing on the convergence of the training of the proposed Fixed and Variable AE frameworks – (i) we analyze the convergence of the proposed AE frameworks by estimating the MI between the input-output of the AE for varying signal-to-noise-ratio (SNR) levels, and (ii) we provide the necessary conditions for AE’s convergence by showing that once the validation CE loss has converged and the NN encoder of AE designs 2^k codewords during the training, the AE frameworks have converged to their maximum potential of decoding the signal.
- To open the black-box of the AE-based BCM and d-BCM designs, we reveal the distinct observations of 2^k codewords designed in $2n$ -dimensional space by the proposed end-to-end training of the Fixed AE framework. Specifically, we utilize the minimum Euclidean distance, packing density, and Kurtosis to understand the designed codewords’ observations at the source node’s NN encoder. The minimum Euclidean distance tells us how far the symbols, representing the codewords, are mapped in the $2n$ -dimensional space. The packing density helps us in analyzing how well the symbols are packed in the $2n$ -dimensional space. The Kurtosis helps us in analyzing the shape of the packed symbols. These metrics reveal interesting observations of the block-coded modulations designed by the AE framework. Further, providing us a way to compare the designed codewords with the conventional coding scheme, opening the black box of the AE framework. Furthermore, we compare the observations of the AE-based BCM and d-BCM designs.

Please note throughout this chapter; we also define some important notations as *Definitions* because they will be used throughout this thesis in the explanation of the AE frameworks.

2.2 Designing AE Framework for the P2P Networks

In Fig. 2.1, we depict an AE framework for the P2P networks that comprises of L fully-connected (dense) layers, wherein the output of the $l^{\text{th}} \in \{1, \dots, L\}$ dense layer $\omega_l \in \mathbb{R}^{\delta_l}$ can be given as

$$\omega_l = \sigma_l(\mathbf{W}_l \mathbf{x}_l + \mathbf{b}_l) \quad (2.2)$$

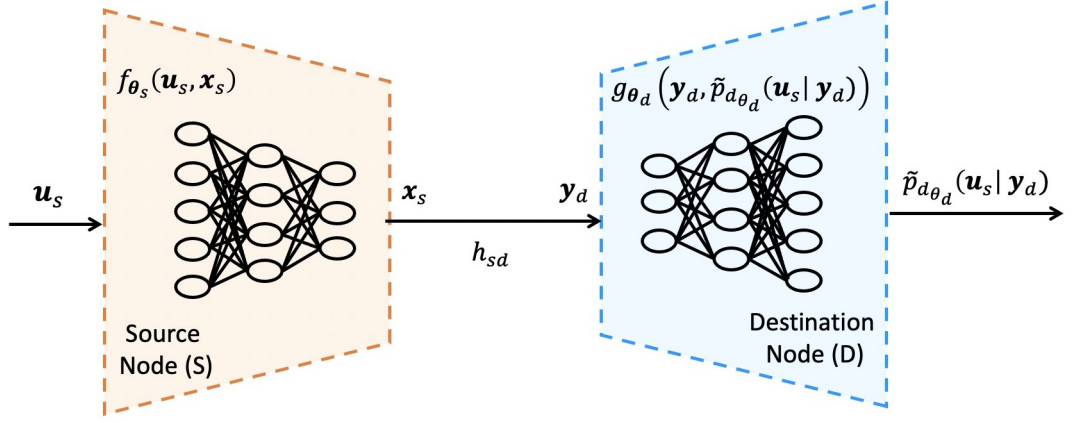


Figure 2.1: An illustration of the DL-based AE frameworks for P2P networks.

where for the the l^{th} dense layer, the ω_l indicates the dense layer's output, $\mathbf{x}_l \in \mathbb{R}^{\delta_l}$ denotes the input, $\mathbf{W}_l \in \mathbb{R}^{\delta_{l-1} \times \delta_l}$ represents the weight matrix between the $(l-1)^{\text{th}}$ and l^{th} dense layers, $\mathbf{b}_l \in \mathbb{R}^{\delta_l}$ is bias vector, and σ_l represents the activation function. For simplicity, we denote all the weight and bias terms of the L layers in the NN using $\theta = \{\mathbf{W}, \mathbf{b}\}$ and refer it as NN's "optimization parameters" collectively.

2.2.1 Designing of NN Encoder

The source node comprises of a NN-based encoder that takes k bits as input $\mathbf{u}_s \in \{0, 1\}^k$ and maps it to n complex baseband symbols $\mathbf{x}_s \in \mathbb{C}^n$. Thereby performing joint block channel-coding and modulation design. In particular, the encoder designs a NN-based mapping function $f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s) : \mathbf{u}_s \rightarrow \mathbf{x}_s$ where θ_s represents the optimization parameters including weights and bias terms of the M dense layers in the NN encoder. Thus, we can define the following

Definition 6 (NN-based mapping function) *The notation $f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s)$ represents an encoder mapping function f at the source node, that takes $\mathbf{u}_s \in [0, 1]^k$ bits as input to give $\mathbf{x}_s \in \mathbb{C}^n$ complex baseband symbols as output using the optimization parameter θ_s .*

Now, we can represent the mapping function at the NN encoder of the source node as

$$f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s) = \mathbf{P}_N(\sigma_M(\mathbf{W}_M \sigma_{M-1}(\mathbf{W}_{M-1} \sigma_{M-2}(\dots \sigma_1(\mathbf{W}_1 \mathbf{u}_s + \mathbf{b}_1) \dots) + \mathbf{b}_{M-1}) + \mathbf{b}_M)) \quad (2.3)$$

wherein, \mathbf{P}_N denotes the power normalization layer, defined as follows:

Definition 7 (Power normalization layer \mathbf{P}_N) *The power normalization layer mandates that the average block power remains n , without employing any trainable NN optimiza-*

tion parameters, given as

$$\mathbf{P}_N(\mathbf{x}) : \|\mathbf{x}\|_2^2 = n \quad (2.4)$$

where \mathbf{x} denotes the $2n$ outputs of the last layer of the NN encoder. Thus, the power normalization layer \mathbf{P}_N ensures $\|f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s)\|_2^2 = n$ or simply $\|\mathbf{x}_s\|_2^2 = n$. Please note that if we remove the power normalization layer, the NN encoder will learn to send the symbols at extremely high power to reduce the reconstruction errors between the input and output of the encoder and decoder, respectively. Thus, it becomes pivotal to include the power normalization layer for an accurate representation of a transmitter node in a communication network.

Also, note that the output of the power normalization layer is the output of the source node's NN encoder, i.e., $\mathbf{x}_s \in \mathbb{R}^{2n}$. As NN can support only real values, thus we have $2n$ real-valued outputs, where $\{1, \dots, n\}$ outputs denote the real part and $\{n+1, \dots, 2n\}$ outputs denote the imaginary part of the n complex baseband symbols.

2.2.2 Signal Transmission

We now perform symbol-by-symbol transmission–reception between the source and destination nodes. For each symbol in the block length (n), the signal received by the destination node is given as

$$y_d = h_{sd} x_s + n_d \quad (2.5)$$

where h_{sd} denotes the Rayleigh block fading (RBF) channel³ between source and destination node with zero mean and unit variance, $h_{sd} \sim \mathcal{CN}(0, 1)$, and n_d is the AWGN at the destination node with zero mean and σ_d^2 variance, $n_d \sim \mathcal{CN}(0, \sigma_d^2)$.

2.2.3 Designing of NN Decoder

Once the n complex baseband symbols are received by the destination node, represented as $\mathbf{y}_d \in \mathbb{C}^n$, it performs block-by-block decoding. In particular, the NN-based decoder at the destination node aims to de-map the received n complex symbols to k soft-probabilistic outputs, denoted by $\tilde{p}_{d\theta_d}(u_s^m | \mathbf{y}_d) \in [0, 1]$, for all $m = \{1, \dots, k\}$. For the sake of clarity, we explicitly define the notation of soft-probabilistic output below:

³In wireless communication, when the channel between the transmitter and receiver remains the same for several symbol intervals, it is referred to as a block fading channel. Moreover, the channel experiences fading effect widely modelled as Rayleigh fading. Thus, a RBF channel is a Rayleigh fading channel that remains the same for a block of symbols.

Definition 8 (Soft-probabilistic outputs notation $\tilde{p}_{d_{\theta_d}}(u_s^m | \mathbf{y}_d)$) For a given block of n complex baseband symbols \mathbf{y}_d , the probability of decoding source node's each input bit u_s^m (for all m) at the NN decoder of the destination node with the optimization parameters θ_d , is represented by $\tilde{p}_{d_{\theta_d}}(u_s^m | \mathbf{y}_d)$, $\forall m$, or, $\tilde{p}_{d_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d)$.

Thus, NN decoder is performing joint channel decoding and demodulation design. In particular, the decoder designs a NN-based de-mapping function $g_{\theta_d}(\mathbf{y}_d, \tilde{p}_{d_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d)) : \mathbf{y}_d \rightarrow \tilde{p}_{d_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d)$, where θ_d represents the trainable parameters including weights and bias terms of the NN-based decoder of $N = L - M$ dense layers in the decoder. Thus, we can define the following

Definition 9 (NN-based de-mapping function) The notation $g_{\theta_d}(\mathbf{y}_d, \tilde{p}_{d_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d))$ represents a decoder de-mapping function g at the destination node, that takes $\mathbf{y}_d \in \mathbb{C}^n$ complex baseband symbols as input to give $\tilde{p}_{d_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d) \in [0, 1]^k$ soft-probabilities as output using the optimization parameter θ_d .

We can represent the de-mapping function at the NN decoder of the destination node as

$$g_{\theta_d}(\mathbf{y}_d, \tilde{p}_{d_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d)) = \sigma_N(\mathbf{W}_N \sigma_{N-1}(\mathbf{W}_{N-1} \sigma_{N-2}(\dots \sigma_1(\mathbf{W}_1 \mathbf{L}_L(\mathbf{y}_d) + \mathbf{b}_1) \dots) + \mathbf{b}_{N-1}) + \mathbf{b}_N) \quad (2.6)$$

where \mathbf{L}_L denotes the *Lambda* layers, defined as:

Definition 10 (Lambda layer \mathbf{L}_L) The *Lambda* layer is a NN layer in the AE framework, which unlike the other layers in the NN, have no NN optimization parameters.

Please note the *Lambda* layer is used to perform signal processing operations, such as performing channel equalization, adding hardware impairments, etc., which can not be done in a NN directly. Further, the power normalization layer, detailed in Definition 7, is a special type of *Lambda* layer that specifically performs the normalization of the signal's block power.

Note the last layer of the NN decoder has a *Sigmoid* activation function that outputs soft-probabilities $\tilde{p}_{d_{\theta_d}}(u_s^m | \mathbf{y}_d)$, $\forall m$, corresponding to the log-likelihood ratios (LLRs) [27], for $\tilde{p}_{d_{\theta_d}}(u_s^m = 1 | \mathbf{y}_d) = \sigma(l_l^m)$, as below

$$\text{LLR}^m := \log \frac{1 - \tilde{p}_{d_{\theta_d}}(u_s^m = 0 | \mathbf{y}_d)}{\tilde{p}_{d_{\theta_d}}(u_s^m = 0 | \mathbf{y}_d)} = l_l^m, \quad \forall m \quad (2.7)$$

These LLRs can be employed directly in an outer code of the powerful long block length decoders, such as LDPC and Turbo codes.

2.3 AE-based BCM and d-BCM Designs

Throughout this thesis, the proposed AE framework performs the BCM designs with the perfect CSI knowledge, and the d-BCM designs without any form of CSI knowledge. Further, in this thesis, we always aim to create a generalizable NN architecture that can perform either BCM or d-BCM designs with minimal changes in the NN architecture. Thus, we employ the CSI knowledge in the Lambda layer L_L of the NN decoders to differentiate in the NN architectures of the AE frameworks for performing BCM or d-BCM designs. Hence, throughout this thesis, only the Lambda layer L_L is modified in the NN architecture for performing BCM and d-BCM design in the AE framework as follows:

- *BCM design* – Herein we assume perfect CSI knowledge at the NN decoder of the destination node. Thus, we directly implement channel h_{sd} equalization in the Lambda layer L_L .
- *d-BCM design* – Herein we assume we do not have any form of CSI knowledge at the NN decoder of the destination node. Thus, we propose to employ a radio transformer network (RTN) in the Lambda layer L_L . The RTN was first introduced in [12], which enables the decoder to decode the signals efficiently in the absence of CSI knowledge. An RTN consists of two parts – (i) a learned parameter estimator $d_\varphi : \mathbb{R}^{2n} \rightarrow \mathbb{R}^z$ which computes a parameter vector $\varphi \in \mathbb{R}^z$ from its input $y_d \in \mathbb{R}^{2n}$, (ii) a layer that combines the output of parameter estimator and input signal, before passing it to the NN decoder.

Since there is no fixed way to determine an optimal RTN architecture for different scenario such as varying rate, users, and nodes in the communication network, we design and develop near-optimal RTN for different scenarios in the rest of the thesis. Moreover, the first part outputs z values; usually, it is of length $2n$ denoting the input signal length. However, we find that depending on the scenario the z output values can be either of length 2 indicating the channel h_{sr} or $2n$ indicating the signal, as will be seen in later Chapters of this thesis. Moreover, the second part that combines the output of the parameter estimator and input signal can either perform concatenation, addition, division, or any other operation depending on the scenario. Thus, designing a suitable RTN for removing the channel impairments is pivotal for d-BCM designs. We further investigate the benefits and limitations of employing an RTN in Chapter 3 of this thesis.

2.4 Training Loss Optimization

The fundamental distinction between the (differential) block-coded modulation design by the proposed AE framework and the conventional networks is that the AE aims to design the block codes using a learning-based approach by updating the NN weights. In contrast, the conventional network aims to employ channel codes such as Hamming codes. Designing optimal block-coded modulations becomes a challenging task in the presence of noise and channel impairments. In fact, it becomes NP-hard to obtain (differential) coded modulation designs for large input-output bits k because the number of codewords increases exponentially as 2^k . Furthermore, determining different (differential) coded modulation designs for different channel characteristics and communication scenarios such as users or nodes in the networks with different interference or noise levels is another significant problem when using the generic conventional codes. For sake of clarity, we define following for, any (n, k) block or rate $R = k/n$ as follows:

Definition 11 (Symbol) *A complex baseband symbol is defined as a complex number indicating the symbol transmitted or received at various nodes in the network.*

Definition 12 (Codeword) *A codeword is a collection of n complex baseband symbols together, representing one of the possible signal point in $2n$ -dimensional space.*

As the input-output to the AE is k bits, thus we can formulate the proposed AE framework as a multi-label binary classification problem, where each of the label represents different bits and each bit can take binary value, that aims to solve the following optimization problem:

$$\begin{aligned} (\boldsymbol{\theta}_s^{\text{AE}}, \boldsymbol{\theta}_d^{\text{AE}}) &= \arg \min_{\boldsymbol{\theta}_s, \boldsymbol{\theta}_d} \mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) \\ &= \arg \min_{\boldsymbol{\theta}_s, \boldsymbol{\theta}_d} \sum_{m=1}^k \mathcal{L}(\boldsymbol{\theta}_s(u_s^m), \boldsymbol{\theta}_d(\boldsymbol{\theta}_s(u_s^m) | \mathbf{y}_d)) \end{aligned} \quad (2.8)$$

Wherein, we utilize the binary CE loss [41] to quantify the de-mapping error at the destination node, given as follows:

Definition 13 (Binary CE loss) *The binary CE loss determines the error between the input bits \mathbf{u}_s at the source node and the bit-wise soft probabilities $\tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s | \mathbf{y}_d)$ decoded at the destination node, given as follows:*

$$\begin{aligned} \mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) &:= \mathcal{L}(\mathbf{u}_s, \tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s | \mathbf{y}_d)) \\ &= \sum_{m=1}^k -(1 - u_s^m) \log_2(\tilde{p}_{d_{\boldsymbol{\theta}_d}}(u_s^m | \mathbf{y}_d)) - u_s^m \log_2(1 - \tilde{p}_{d_{\boldsymbol{\theta}_d}}(u_s^m | \mathbf{y}_d)) \end{aligned} \quad (2.9)$$

A similar optimization problem occurs in the conventional networks (CN) where encoder maps the k bits to 2^k possible codewords and a decoder that de-maps the n received symbols \mathbf{y}_d to obtain intended k bits, as shown below,

$$\begin{aligned} (\boldsymbol{\theta}_s^{\text{CN}}, \boldsymbol{\theta}_d^{\text{CN}}) &= \arg \min_{\boldsymbol{\theta}_s, \boldsymbol{\theta}_d} \pi(n, k) \\ &= \arg \min_{\boldsymbol{\theta}_s, \boldsymbol{\theta}_d} \mathcal{P}[\boldsymbol{\theta}_d(\mathbf{y}_d) \neq u_s^m | \boldsymbol{\theta}_e(u_s^m), \mathbf{y}_d], \forall m \end{aligned} \quad (2.10)$$

Thus, inspired by the *Theorem 1* in [15] for symbol-wise AE P2P networks that only considers coded-modulation designs while bit-labelling has to be performed heuristically solving a $2^k!$ combinatorial problem. We propose the following theorem which additionally deals with automatic bit-labelling with coded-modulation design for bit-wise AE-based networks.

Remark 1 *The optimization problems of the proposed AE framework in (2.8) and the communication networks in (2.10) for the P2P network obtain the same encoder-decoder and bit-labelling when the AE is trained with bits as input-output, sigmoid activation at the last layer of the decoder and binary CE loss is optimized.*

Proof: Please refer Appendix 2.B.

Thus, the *Remark 1* states the existence of an AE framework for the P2P networks that can be employed to obtain optimum channel codes for any block length with optimum bit-labelling. Until now no tool was known that can obtain optimum block codes with automatic bit-labelling as a result of mathematical modeling of the communication system. In fact, the Shannon's coding theorem only states the existence of a good code without specificity, and only for infinite block lengths [41], [15]. Thus, proposed AE framework can in principle be used for any P2P scenarios to determine the optimum error correction codes with automatic bit-labelling, even for short block length.

For sake of completion, similar to [38], we can show that the generalized mutual information (GMI) of a conventional networks can be given as

$$\text{GMI} = 1 - \mathbb{E}_{\mathbf{u}_s, \mathbf{l}_1} [\log_2 (1 + \exp(-(-1)^{\mathbf{u}_s} \mathbf{l}_1))] \quad (2.11)$$

where $\mathbf{l}_1 \in \mathbb{R}^k$ denotes the LLRs in (5.13). Rearranging the binary CE loss in (2.9), we get

$$\mathcal{L}(\mathbf{u}_s, \tilde{p}_{d_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d)) = \mathbb{E}_{\mathbf{u}_s, \mathbf{L}} [\log_2 (1 + \exp(-(-1)^{\mathbf{u}_s} \mathbf{L}))] \quad (2.12)$$

where $\mathbf{L} \in \mathbb{R}^k$ denotes the output of the AE's decoder prior to applying the sigmoid activation function, i.e. $\tilde{p}_{d_{\theta_d}}(u_s^m | \mathbf{y}_d) = \frac{1}{1+e^{-L^m}}, \forall m$ [41]. Thus, equating (2.11) and

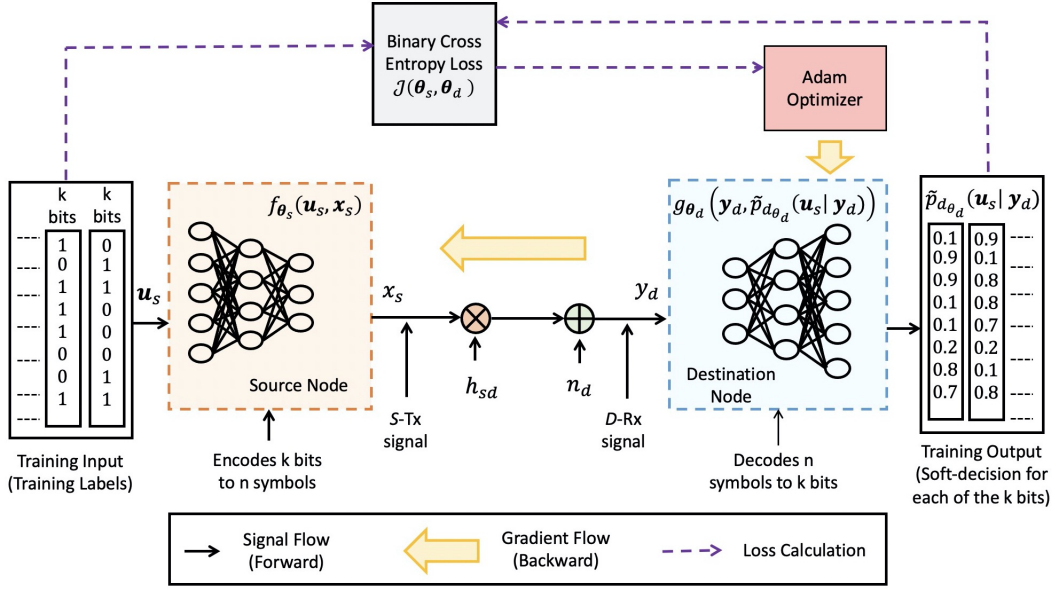


Figure 2.2: Illustration of the training of the proposed AE framework.

(2.12), we have

$$\mathcal{L}(\mathbf{u}_s, \tilde{p}_{d\theta_d}(\mathbf{u}_s|\mathbf{y}_d)) = 1 - \text{GMI} \quad (2.13)$$

This shows that minimizing the binary CE loss also leads to the maximization of GMI [38].

2.5 Training and Testing Procedure

In this section, we detail the training and testing procedure of the AE framework. Let us consider a dataset $\mathbf{Z} \in [0, 1]^{S \times k}$ with S total samples divided as $\{S_{\text{train}}, S_{\text{test}}\}$ training and testing samples, where each sample contains k bits of data.

2.5.1 Training Procedure

We train the proposed AE framework by estimating the expected loss in (2.9) using the mini-batch training [41] as follows:

$$\mathcal{J}(\theta_s, \theta_d) = \frac{1}{B^s} \sum_{q=1}^Q \sum_{b^s=1}^{B^s} \sum_{m=1}^k \mathcal{L}(u_s^{(q \times b^s, m)}, \tilde{p}_{d\theta_d}(u_s^{(q \times b^s, m)} | \mathbf{y}_d^{(q \times b^s)})) \quad (2.14)$$

where B^s denotes the batch size and $Q = S_{\text{train}}/B^s$ denoting the number of samples in a batch. The weights and bias terms in the NN-based AE framework is updated using the Adam optimizer via the back-propagation method [41]. Let us denote all the NN optimization parameters as $\Theta = \{\theta_s, \theta_d\}$, then the update rule can be given as follows. We update the weights using the Adam optimizer [42] as detailed below. Let η be the

learning rate, $\beta_1, \beta_2 \in [0, 1)$ be the exponential decay rates for the moment estimates, $\mathcal{J}(\Theta)$ be the stochastic objective function with the NN parameters Θ , Θ_0 be the initial NN parameter, $m_0 = 0$ be the first moment vector, $v_0 = 0$ be the second moment vector, and t be the time step. Then, the Adam optimizer implements the following until the NN parameter Θ_t is converged [42]:

- Update the time step: $t \leftarrow t + 1$.
- Get the gradients w.r.t. to the loss function at time step t : $g_t \leftarrow \nabla_{\Theta} \mathcal{J}(\Theta_{t-1})$.
- Update biased first moment estimate: $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$.
- Update biased second raw moment estimate: $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$.
- Determine the bias-corrected first moment estimate: $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$.
- Determine the bias-corrected second raw moment estimate: $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$.
- Update the parameters: $\Theta_t \leftarrow \Theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$.

We summarize the training process of the proposed AE framework in Fig. 6.4.1, that aims to maximize the chances of reconstruction of the intended signal \mathbf{u}_s at the destination node by learning the NN optimization parameter Θ .

2.5.2 Testing Procedure

For each $r^{\text{th}} = \{1, \dots, S_{\text{test}}\}$ testing sample, we employ the trained AE framework with optimization parameter Θ and obtain the soft probabilistic outputs $\tilde{p}_{d_{\theta_d}} \left(u_s^{(r,m)} | \mathbf{y}_d^{(r)} \right)$, $\forall r, m$. While, the output bits at the destination node can be predicted as $\hat{u}_s^{(r,m)}$ by keeping a hard-decision threshold such as 0.5 on the $\tilde{p}_{d_{\theta_d}} \left(u_s^{(r,m)} | \mathbf{y}_d^{(r)} \right)$, as follows:

$$\hat{u}_s^{(r,m)} = \begin{cases} 0, & \text{if } \tilde{p}_{d_{\theta_d}} \left(u_s^{(r,m)} | \mathbf{y}_d^{(r)} \right) < 0.5, \\ 1, & \text{if } \tilde{p}_{d_{\theta_d}} \left(u_s^{(r,m)} | \mathbf{y}_d^{(r)} \right) \geq 0.5, \end{cases} \quad \forall r, m. \quad (2.15)$$

2.6 Variable AE versus Fixed AE

In this section, we design and evaluate the Variable and Fixed AE frameworks.

2.6.1 Designing of Variable and Fixed AE Frameworks

Now, we detail various aspects of designing and developing the Variable and Fixed AE frameworks below:

Table 2.2: NN architecture of *Variable AE*.

Node	Layer No. (l)	Nodes (δ_l)	Remarks
Encoder	$l = 0$	k	Input (\mathbf{x})
	$l = 1$	2^k	$\sigma_1 = \text{Tanh}$
	$l = 2$	$2n$	$\sigma_2 = \text{Linear}$
	$l = 3$	$2n$	Power Normalization (\mathbf{P}_N)
	$l = 4$	$2n$	Output (\mathbf{x}_s)
Channel	$l = 5$	$2n$	RBF channel
Noise	$l = 6$	$2n$	AWGN
Decoder	$l = 7$	$2n$	Input (\mathbf{y}_d)
	$l = 8$	$2n$	Lambda layer \mathbf{L}_L
	$l = 9$	2^k	$\sigma_9 = \text{Tanh}$
	$l = 10$	k	$\sigma_{10} = \text{Sigmoid}$
	$l = 11$	k	Output ($\hat{\mathbf{x}}$)

 Table 2.3: NN architecture of *Fixed AE*.

Node	Layer No. (l)	Nodes (δ_l)	Remarks
Encoder	$l = 0$	k	Input (\mathbf{x})
	$l = 1$	2^8	$\sigma_1 = \text{Tanh}$
	$l = 2$	2^7	$\sigma_2 = \text{Tanh}$
	$l = 3$	2^6	$\sigma_3 = \text{Tanh}$
	$l = 4$	$2n$	$\sigma_4 = \text{Linear}$
	$l = 5$	$2n$	Power Normalization (\mathbf{P}_N)
	$l = 6$	$2n$	Output (\mathbf{x}_s)
Channel	$l = 7$	$2n$	RBF channel
Noise	$l = 8$	$2n$	AWGN
Decoder	$l = 9$	$2n$	Input (\mathbf{y}_d)
	$l = 10$	$2n$	Lambda layer \mathbf{L}_L
	$l = 11$	2^9	$\sigma_{11} = \text{Tanh}$
	$l = 12$	2^8	$\sigma_{12} = \text{Tanh}$
	$l = 13$	2^6	$\sigma_{13} = \text{Tanh}$
	$l = 14$	k	$\sigma_{14} = \text{Sigmoid}$
	$l = 15$	k	Output ($\hat{\mathbf{x}}$)

Table 2.4: NN architecture of RTN.

Node	Layer No. (l)	Nodes (δ_l)	Remarks
RTN	$l = 0$	$2n$	Input (\mathbf{y}_d)
	$l = 1$	2^8	$\sigma_1 = \text{Tanh}$
	$l = 2$	2^6	$\sigma_3 = \text{Tanh}$
	$l = 3$	$2n$	$\sigma_3 = \text{Linear}$
	$l = 4$	$[2n, 2n]$	Concatenate output of $l = 3$ and \mathbf{y}_d

Table 2.5: Training hyper-parameters setting.

Parameters	BCM w/ CSI	d-BCM w/o CSI
Batch-size (B)		6000
Epochs (E)	300	200
Optimizer		Adam
Initializer		Glorot
Training E_b/N_0 (from set \mathcal{S})		[3, 8, 13, 25] dB
Testing E_b/N_0		-4 dB to 20 dB
Blocks of Training Samples per E_b/N_0		3×10^4
Blocks of Testing Samples per E_b/N_0		10^5
Parameters for step-decay learning-rate		
Initial learning-rate (τ_0)		10^{-3}
Drop (η)		0.5
Drop epochs (D_E)	30	20
Minimum learning-rate (τ_{\min})		10^{-5}
Parameters for early-stopping		
Monitor		Validation loss
Minimum change		10^{-4}
Patience		10

- NN Architectures** – Firstly, we detail the NN architecture for the Variable AE and Fixed AE in Table 2.2 and Table 2.3, respectively. As can be seen, in Fixed AE the number of neurons is independent of k , unlike the Variable AE framework. Now, we detail about the Lambda layers \mathbf{L}_L ($l = 8$ in variable AE and $l = 10$ in Fixed AE). As discussed in Sec. 2.3, in case of BCM we perform channel equalization in the Lambda layers \mathbf{L}_L and in case of d-BCM we apply an RTN in the Lambda layers \mathbf{L}_L . We detail the NN architecture for the RTN in Table 2.4. Please note we use same RTN in both the Fixed and Variable AE frameworks for d-BCM design.
- Training and Testing Dataset Creation** – Throughout this thesis, we aim to create an AE framework that remains unaffected of the testing E_b/N_0 , where E_b/N_0 denotes energy per bit E_b to noise spectral density N_0 ratio. Throughout this thesis, we create the training and testing dataset, comprising of transmitted bits, fading channels, and noise with different E_b/N_0 , in the MATLAB. While we implement the AE framework in python using Keras [43] with TensorFlow [44] as backend. We train the AE framework using the training dataset created in MATLAB. Specifically, we create a single training dataset of multiple E_b/N_0 values (represented by set \mathcal{S}) and train a single AE framework on the same. Then, we test the trained AE framework on various testing E_b/N_0 . We report the training and testing E_b/N_0 in Table 2.5.

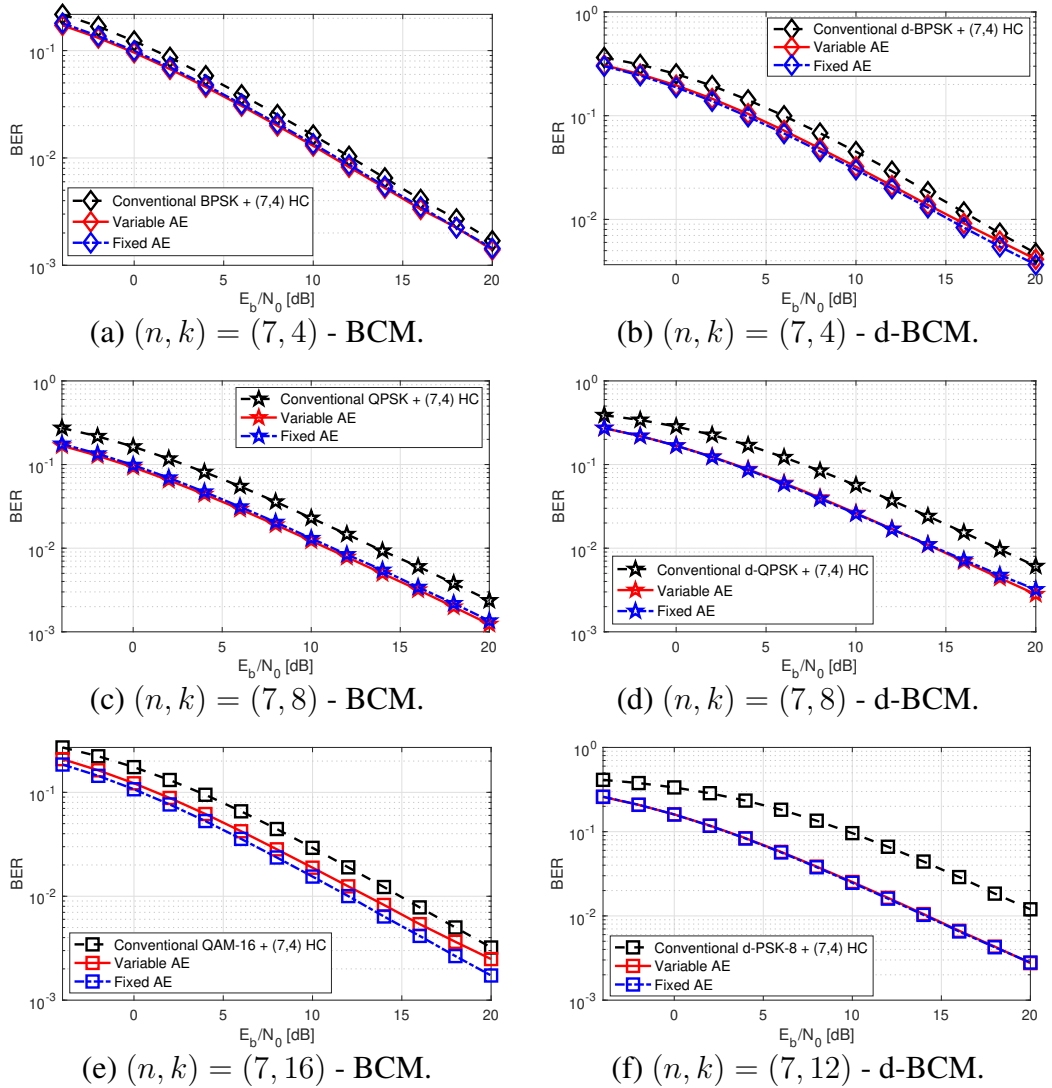


Figure 2.3: Performance evaluation for Fixed AE and Variable AE for varying rates (R). In Fig. (a), (c), (e) we analyze BCM designs and in Fig. (b), (d), (f) we analyze d-BCM designs.

- *Hyper-Parameter Setting* – To train the proposed AE frameworks we utilize SGD with Adam optimizer [42], where the weights are initialized using Glorot initializer [45]. We utilize step-decay⁴ method to update the learning rate. In particular, we start with an initial learning-rate τ_0 for the first epoch and then drop the learning rate by η after every D_E epochs (we also constraint the minimum learning rate to be τ_{\min}). We also utilize early stopping [46] to stop the training of the proposed AE framework if no significant update is made to the validation loss during the training process. We summarize all the training hyper-parameter settings in Table 2.5, that remains same for both the Fixed and Variable AE frameworks, for any (n, k) or rate R . Please note that in case of $(n, k) = (7, 16)$ Variable AE framework,

⁴Step Decay is a learning rate schedule that drops the learning rate by a factor every few epochs, where the number of epochs is a hyperparameter.

we note that it is best to train the Variable AE framework without updating the learning rate using step-decay and without early stopping. Specifically, we train the $(n, k) = (7, 16)$ Variable AE framework for 100 epochs with 256 batch size and a fixed 10^{-3} learning-rate.

2.6.2 Numerical Results

With the advent of internet-of-things (IoT) and ultra-reliable low-latency communication (URLLC), reducing energy consumption while satisfying the low-latency requirements has become pivotal. Thus, the employment of short block codes has gained considerable traction in industrial applications. For example, recent works [47] propose to employ shorter codes with block size of at most 128 sizes. However, conventional codes below this block size are not able to produce effective coding gains for reliable communications. Moreover, even shorter block sizes are required for meeting the ultra-reliability constraints of the upcoming URLLC networks, especially for the wireless sensor nodes. Further, recent studies [48] show that apart from polar codes, the generalized low-density parity-check (G-LDPC) codes are a much stronger candidate for the URLLC networks. In contrast to the LDPC codes, the G-LDPC codes replace the single parity-check bits with $(7,4)$ Hamming codes. Thus, we focus on $(7,4)$ Hamming codes as the baseline for the proposed AE framework. Further, another advantage of considering such small block codes is that we can consider an outer powerful code such as LDPC with AE-based designed codes for $(7,4)$ as the inner code, and we can achieve similar or greater performance gains. A similar technique is employed in satellite communications for improving decoding performance. Please see Sec. 4.5.9 for further details. Thus, designing codes for smaller block lengths provide us with the flexibility to use these codes directly for small sensor devices, etc, with limited processing capabilities, or, be utilized in the powerful codes such as G-LDPC, or, be combined with outer powerful codes for designing longer block codes for wireless communications.

Throughout this chapter, we consider RBF channels, such that it remains constant for a transmission block $n = 7$ and changes thereafter. We consider AWGN at the destination node with fixed variance $\sigma^2 = (2RE_b/N_0)^{-1}$. Moreover, for the conventional scenario, we utilize traditional modulation techniques and $(7, 4)$ Hamming code with the maximum likelihood decoding (MLD). Also, we utilize the notation (n, k) to denote k bits and n block length, but also for representing rate $R = k/n$.

Now, we evaluate the performance of the proposed Fixed AE and Variable AE frameworks for the scenarios – BCM for $(n, k) = \{(7, 4), (7, 8), (7, 16)\}$ and d-BCM for $(n, k) = \{(7, 4), (7, 8), (7, 12)\}$. For the sake of fair comparison, we utilize BPSK for $(n, k) = (7, 4)$, QPSK for $(n, k) = (7, 8)$, PSK-8 for $(n, k) = (7, 12)$, and QAM-16 for $(n, k) = (7, 16)$ in conventional scenarios. We show differential modulation schemes by

Table 2.6: Number of optimized parameters and time-cost analysis. Training and testing time is shown in seconds. Testing time is shown for a total of 10^5 blocks of (n, k) data. The number of optimized parameters in RTN is 21, 198.

(n, k)	AE	CSI knowl.	No. of opt. parameters	Training time (in sec.)	Testing time (in sec.)
(7, 4)	Fixed	✓	199, 058	25.53	4.61
	Variable	✓	626	32.62	4.26
(7, 8)	Fixed	✓	200, 342	35.19	4.50
	Variable	✓	11, 798	29.07	4.31
(7, 16)	Fixed	✓	202, 910	32.44	4.54
	Variable	✓	4, 063, 262	406.16	4.77
(7, 4)	Fixed	✗	227, 424	24.22	3.77
	Variable	✗	22, 048	20.31	3.41
(7, 8)	Fixed	✗	228, 708	27.28	3.79
	Variable	✗	36, 580	16.69	3.47
(7, 12)	Fixed	✗	229, 992	28.20	3.65
	Variable	✗	299, 752	12.33	3.50

using (d-), such as d-BPSK, d-QPSK and d-PSK-8.

- *BER Analysis* – Firstly, in Fig. 2.3, we can see that the proposed Fixed AE and Variable AE framework outperforms the conventional modulation with $(7, 4)$ Hamming code technique for all the E_b/N_0 and varying rates or (n, k) . This suggests the advantages of using AE framework over the conventional methods. Also, we can see that BER performance gains of the AE increases with the rate R . Besides, BER performance gains of the AE-based d-BCM design is greater than AE-based BCM design. This shows that RTN is helping in designing differential coded-modulations by learning the impairments due to RBF channels effectively. Secondly, in Fig. 2.3e, for $(n, k) = (7, 16)$, we note that Fixed AE outperforms the Variable AE framework. This is because the Variable AE framework has a single dense layer with 2^{16} neurons in the encoder and decoder, which are a humongous amount of neurons, whereas the Fixed AE has small number of neurons in multiple dense layers. Thereby the number of parameters that needs to be optimized for Variable AE is 4, 063, 262 and for Fixed AE is 202, 910. Thus, as the size of k increases to a large value (greater than 12) it becomes better both in terms of BER and optimizing parameters to have multiple dense layers with less neurons than to have a single dense layer with 2^k neurons.
- *Number of Optimized Parameters* – We can see in Table 2.6 that number of optimized parameters remains $\approx 200, 000$ for Fixed AE for any (n, k) but the number of optimized parameters for Variable AE increases exponentially with input-output bits k . Thus, it is cost-effective to employ Variable AEs for smaller input-output bits $k \leq 12$ but it becomes cost-effective to employ Fixed AEs for larger input-output

bits $k > 12$.

- *Time-Cost Analysis* (in Table 2.6) – Since we are using early stopping to stop the training once the validation CE loss has converged, instead of fixed number of epochs for training, the training time does not directly depend of the number of optimized parameters. However, in case of Variable AE for (7, 16) the number of optimizing parameters are so high it leads to $10\times$ training time compared to Fixed AE for (7, 16). Further, testing time is more for BCM design compared to d-BCM design because in scenarios with CSI knowledge we utilize Lambda layers in the decoders for channel equalization whereas we use RTN in scenarios without CSI knowledge. Thus, RTN is faster than Lambda layers making the testing time lower for AE frameworks without CSI knowledge.

Thus, we can conclude that Variable AE is good for low rate $R \leq 8/7$, however when the rate becomes high $R > 8/7$ then it becomes advantageous to employ the Fixed AE as the number of optimization parameters in the AE reduces, but also for higher rates $R > 12/7$ in terms of the BER performance. Also, the BER performance gains of the AE increases with the rate R . Besides, BER performance gains of the AE frameworks without the CSI knowledge is greater than AE frameworks with the CSI knowledge, compared to the conventional networks.

Also, Fixed AE enables us to design a single NN architecture for AE frameworks that can be employed for varying rates R [bits/channel-reuse]. In practical deployment, the re-training of the AE frameworks with varying environmental conditions is inevitable. In such scenarios, employing a Fixed AE framework becomes advantageous because we have a single fixed weight matrix size at the encoder and decoder that needs to be updated.

2.7 Information-Theoretic View of Convergence of the Proposed AE Frameworks

In this section, we demonstrate the impact of varying E_b/N_0 levels and designing of BCM/d-BCM designs in the presence/absence of CSI knowledge on the convergence of the training (detailed in Sec. 2.5.1) of the proposed Fixed and Variable AE frameworks, by giving the following Remarks.

Remark 2 *For a sufficiently large block length (n), the training of Fixed AE framework converges to a global minima above a minimum required E_b/N_0 .*

Remark 3 *For a sufficiently large block length (n), the training of Fixed AE framework converges faster than the Variable AE framework.*

Remark 4 *The CSI knowledge is helpful to reach near to the global minima faster, i.e. the convergence of the training of AE performing BCM designs with CSI knowledge is faster than AE performing d-BCM designs without CSI knowledge.*

Proof: For any received block of (n) complex baseband symbols, \mathbf{y}_d , the binary CE loss in (2.9) can also be represented as

$$\mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) = H\left(p_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|\mathbf{y}_d), \tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|\mathbf{y}_d)\right) \quad (2.16)$$

where $H\left(p_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|\mathbf{y}_d), \tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|\mathbf{y}_d)\right)$ denotes the binary CE loss between the true distribution $p_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|\mathbf{y}_d)$ at the NN encoder of the source node and the learnt distribution $\tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|\mathbf{y}_d)$ at the NN decoder of the destination node, with parameters $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_d$, respectively. Now, we firstly derive the binary CE loss by taking the expectation of (2.16) with respect to \mathbf{y}_d , as follows:

$$\begin{aligned} \mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) & \stackrel{(a)}{=} \mathbb{E}_{\mathbf{y}_d} \left[H\left(p_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|\mathbf{y}_d), \tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|\mathbf{y}_d)\right) \right] \\ & \stackrel{(b)}{=} -\mathbb{E}_{\mathbf{y}_d} \left[\sum_{m=1}^k p_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s^m|\mathbf{y}_d) \log \tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s^m|\mathbf{y}_d) \right] \\ & \stackrel{(c)}{=} -\sum_{m=1}^k \int_{\mathbf{y}_d} p(\mathbf{y}_d) p_{s_{\boldsymbol{\theta}_s}}(u_s^m|\mathbf{y}_d) \log \tilde{p}_{d_{\boldsymbol{\theta}_d}}(u_s^m|\mathbf{y}_d) d\mathbf{y}_d \\ & \stackrel{(d)}{=} \sum_{m=1}^k \int_{\mathbf{y}_d} p_{s_{\boldsymbol{\theta}_s}}(u_s^m, \mathbf{y}_d) \log \left[\frac{p_{s_{\boldsymbol{\theta}_s}}(u_s^m|\mathbf{y}_d)}{p_{s_{\boldsymbol{\theta}_s}}(u_s^m|\mathbf{y}_d) \tilde{p}_{d_{\boldsymbol{\theta}_d}}(u_s^m|\mathbf{y}_d)} \right] d\mathbf{y}_d \\ & \stackrel{(e)}{=} \sum_{m=1}^k \int_{\mathbf{y}_d} p_{s_{\boldsymbol{\theta}_s}}(u_s^m, \mathbf{y}_d) \log \left[\frac{p_{s_{\boldsymbol{\theta}_s}}(u_s^m|\mathbf{y}_d)}{\tilde{p}_{d_{\boldsymbol{\theta}_d}}(u_s^m|\mathbf{y}_d)} \right] d\mathbf{y}_d \\ & \quad - \sum_{m=1}^k \int_{\mathbf{y}_d} p_{s_{\boldsymbol{\theta}_s}}(u_s^m, \mathbf{y}_d) \log p_{s_{\boldsymbol{\theta}_s}}(u_s^m, \mathbf{y}_d) d\mathbf{y}_d \\ & \stackrel{(f)}{=} D_{\text{KL}}\left(p_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|\mathbf{y}_d) \parallel \tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|\mathbf{y}_d)\right) + H_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|\mathbf{y}_d) \\ & \stackrel{(g)}{=} D_{\text{KL}}\left(p_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|\mathbf{y}_d) \parallel \tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|\mathbf{y}_d)\right) + H(\mathbf{u}_s) - I_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s; \mathbf{y}_d) \end{aligned} \quad (2.17)$$

where $D_{\text{KL}}\left(p_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|\mathbf{y}_d) \parallel \tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|\mathbf{y}_d)\right)$ denotes the Kullback-Leibler (KL)-divergence loss between the true distribution $p_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|\mathbf{y}_d)$ at the NN encoder of the source node and the learnt distribution $\tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|\mathbf{y}_d)$ at the NN decoder of the destination node, with parameters $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_d$, respectively, $H(\mathbf{u}_s)$ denotes entropy of the input bits \mathbf{u}_s at the source node, and $I_{s_{\boldsymbol{\theta}_s}}(\mathbf{u}_s; \mathbf{y}_d)$ is the MI between the input bits \mathbf{u}_s and the received signal at the destination node \mathbf{y}_d with parameters $\boldsymbol{\theta}_s$.

Moreover, the (2.17) is derived using the following steps: (a) is obtained by the taking expectation over \mathbf{y}_d , (b) comes from the definition of binary CE loss, in (c) we open

the expectation, in (d) we multiply and divide by $p_{s_{\theta_s}}(u_s^m, \mathbf{y}_d)$, in (e) we open the log function, (f) comes from the definition of KL divergence loss and conditional entropy, and (g) utilizes the identity $H(a|b) = H(a) - I(a; b)$ [49].

We can now obtain the estimated MI [36], which we define as follows:

Definition 14 (Estimated MI) *The estimated MI, represented by (\mathcal{I}), is defined as the MI between the input bits and received symbols, subtracted by the relative entropy between the learnt distributions at the NN encoder and decoder pair:*

According to this definition, the estimated MI can be given as follows:

$$\mathcal{I} := I_{s_{\theta_s}}(\mathbf{u}_s; \mathbf{y}_d) - D_{\text{KL}}\left(p_{s_{\theta_s}}(\mathbf{u}_d|\mathbf{y}_d) \parallel \tilde{p}_{d_{\theta_d}}(\hat{\mathbf{u}}_s|\mathbf{y}_d)\right) \quad (2.18)$$

Now, using (2.17), the estimated MI in (2.18) becomes

$$\mathcal{I} := H(\mathbf{u}_s) - \mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) \quad (2.19)$$

Since the first term on R.H.S, $H(\mathbf{u}_s)$ in (2.19), remains a constant, thus the changes in the estimated MI in (2.18) only depends on the binary CE loss term $\mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d)$.

Lastly, by simulations, we analyze the convergence of the training of proposed Fixed and Variable AE frameworks. In particular, we train a separate Fixed (or Variable) AE for each E_b/N_0 (in dB) level using the NN architecture and hyper-parameter settings as described in Sec. 2.6.1. Once the Fixed (or Variable) AE is trained we note the validation CE loss ($\mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d)$) at the last epoch before early stopping the training, and obtain the estimated MI (\mathcal{I}) as described in (2.19).

In Fig. 2.4, we show the estimated MI for the BCM and d-BCM designs using the Fixed and Variable AE frameworks for varying rates $R = k/n$ [bits/channel reuse]. Also, we keep block size $n = 7$ to have sufficiently large block length for encoding of up to 16 bits. Please note that we will discuss in detail about block length in next Section.

In Fig. 2.4, we can see that as the E_b/N_0 (in dB) increases the estimated MI (\mathcal{I}) increases for the BCM and d-BCM designs by the Fixed and Variable AE frameworks. This is because the binary CE loss reduces while training at higher E_b/N_0 . Moreover, we can see that in Fig. 2.4, very evidently in Fig. 2.4c, the estimated MI of the Fixed AE increases until it reaches the upper bound of k , while the estimated MI of the Variable AE is either reaching the upper bound of k slower than Fixed AE or is unable to reach the upper bound of k . Directly from (2.19), it suggests that KL-divergence loss approaches 0 making $I_{s_{\theta_s}}(\mathbf{u}_s; \mathbf{y}_d) = H(\mathbf{u}_s)$ for the Fixed AE, and sometimes for the Variable AE. Thus, the training of Fixed AE-based P2P networks converges near to a global minima above a minimum required E_b/N_0 , and also the training convergence of Fixed AE is much better than the Variable AE. In Fig. 2.4, we can also see that the BCM designs with the

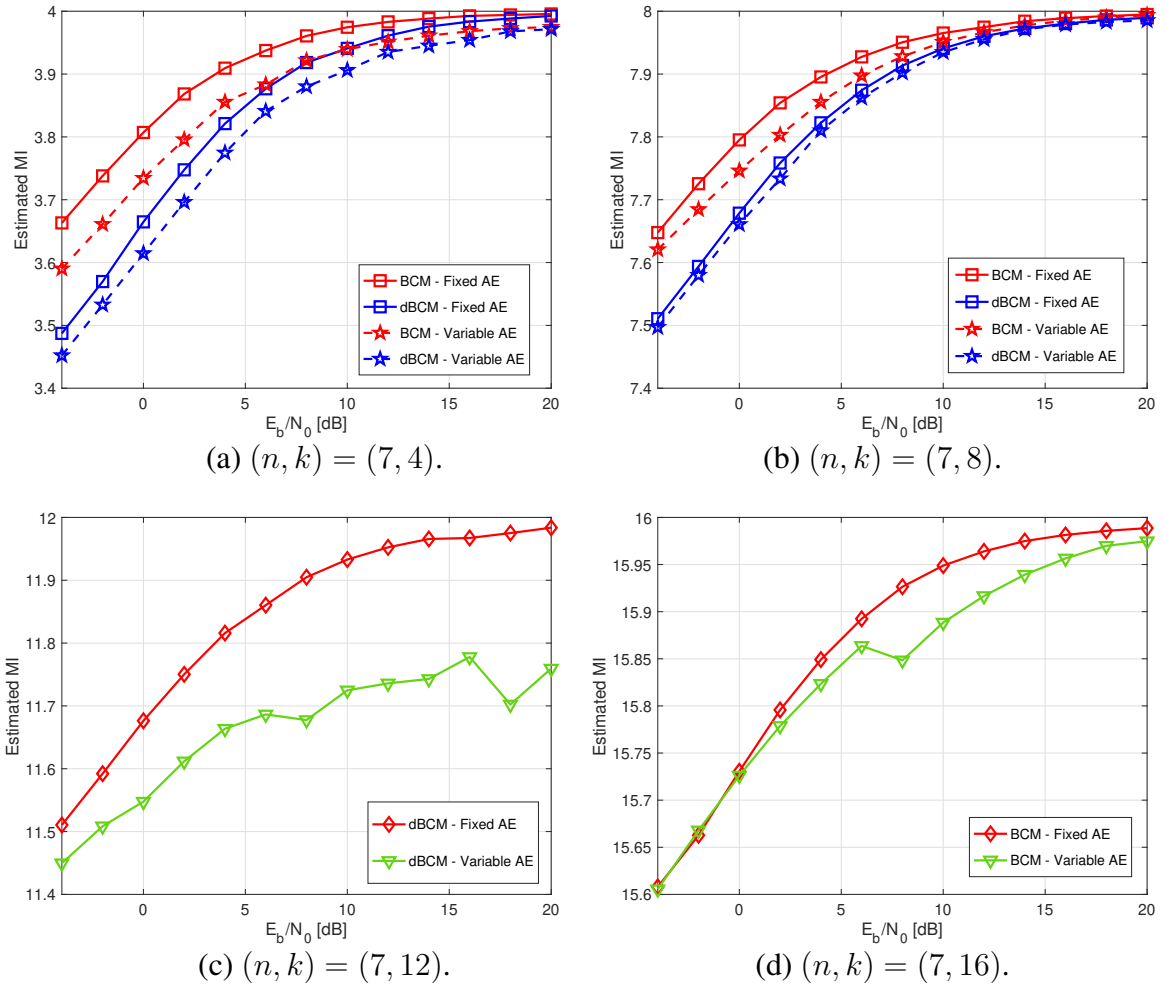


Figure 2.4: Estimated MI for BCM and d-BCM designs using the Fixed and Variable AE frameworks for varying rates R .

CSI knowledge converge to the upper bound of k at a lower E_b/N_0 compared to the d-BCM designs without the CSI knowledge for both the Fixed and Variable AE frameworks. Thus, CSI knowledge is helpful to reach near to the global minima faster. It is important to note that we can't find the global minima of the NN parameters with respect to the binary CE loss. But, surprisingly we don't need to find the global minima. Empirically, the authors in [50, 51] found that despite the non-convexity, the local minima's are rare and they are all very similar to each other and the global minima. Interested readers, please refer to the theoretical insights presented in [50, 51].

■

2.8 Necessary Conditions for AE Framework’s Convergence

The process of training the NN-based AE frameworks include determining hyper parameter settings [41], such as weight initialization, activation functions, learning-rate, batch size, etc, for example as also performed in Section 2.5.1. Although the training process of NNs have seen advancements, yet no universal technique exists to optimize these hyper-parameter settings, leading to suboptimal choices, forcing the AE to get stuck in a local minima while minimizing the CE loss in (2.8) (c.f. [52, 53]). Thus, in existing literature [12]–[40], hyper-parameter settings are obtained sub-optimally by hit-and-trial method. Which includes training an AE framework with various hyper-parameter settings, monitoring the validation⁵ CE loss and picking the hyper-parameter settings that give the minimum validation CE loss during the training. However from coded modulation perspective, a major problem with determining the AE’s convergence by simply monitoring the validation CE loss is that the validation CE loss for most of the non-optimal hyper-parameter settings also reduces with training epochs and thus we can not surely determine if the AE performing the BCM and d-BCM designs are converged by only monitoring the validation CE loss. Thus, *for any given hyper-parameter settings, we need to determine the relationship between the BCM or d-BCM designs performed by the AE, which can indicate if the trained AE has converged to its maximum potential of decoding.*

Remark 5 *For any given hyper-parameter settings and rate $R = k/n$, with sufficiently large block length (n), the two necessary conditions for the convergence of training of the AE frameworks performing BCM or d-BCM designs are detailed as follows:*

C_1 : *The validation CE loss of the AE framework has converged.*

C_2 : *The NN encoder of the AE designs 2^k codewords.*

Remark 6 *For any given hyper-parameter settings and rate $R = k/n$, with sufficiently large block length (n), the BER performance of the AE framework performing BCM design converge with at least 2^{k+1} training samples, while that for d-BCM design converge with at least 2^{k+5} training samples.*

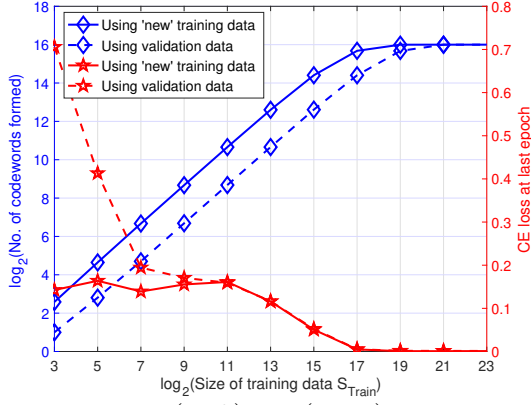
Proof: In this work, we are solving a bit-decoding problem formulated as a multi-label classification problem by minimizing the binary CE loss of the designed AE framework.

⁵Typically, a training dataset is divided into a 4 : 1 ratio of ‘new’ training set and validation set. Where NN is trained on the ‘new’ training set and tested on a validation set, during training, to gauge the NN’s potential testing performance in future. The CE loss calculated on new training set is referred to as ‘training CE loss’ and the CE loss calculate on validation set is referred to as ‘validation CE loss’.

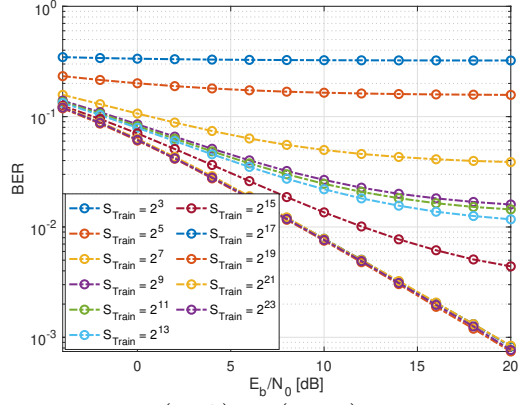
The AE frameworks are based on neural networks (NNs). A general method of training the NN-based architectures is to monitor the validation loss, once it is converged, we can say the NN is trained. Thus, we can say that the training of the AE framework is converged when the validation CE loss cannot be improved further. For any given AE framework with fixed hyper-parameter settings, the convergence of the validation CE loss depends on two factors – (1) if the validation CE loss is converged concerning the training epochs, i.e., training the AE for a larger number of epochs does not lead to any improvement in minimizing the validation CE loss, and (2) if the validation CE loss is converged concerning the training dataset size, i.e., training the AE for a larger training dataset does not lead to any improvement in minimizing the validation CE loss. Thus, the C_1 in *Remark 4* has two parts. Firstly, training of the AE with respect to the number of epochs, for a given training samples. A NN-based model is defined to be *underfitting* the training data when the model performs poorly on the training data, whereas *overfitting* the training data when the model performs well on the training data but does not perform well on the validation data. There are well-known techniques like *early stopping*, which we utilized in the training of the AE (detailed in Sec. 2.5.1), to stop the training of an AE once the validation CE loss starts to increase, as any more training of the AE will reduce its generalizability. This is because the early stopping on the gradient descent creates generalizable NN frameworks, that also remains robust to corrupted labels [46, Theorem 2.2]. Thus the first part of C_1 is easily satisfied [46]. Secondly, training of AE with respect to number of training samples. Since the proposed AE learns the BCM and d-BCM design in the presence of fading channels and noise, thus the AE framework must be trained with enough samples to be generalizable in the future testing phase. Thus, second part of C_1 states that the training of the proposed AE is converged if increasing the training samples does not help in further reducing the validation CE loss of the AE frameworks.

Now, we focus on C_2 in *Remark 4*. In this chapter, we model the problem of designing the BCM and/or d-BCM using the AE frameworks. The proposed AE framework is modelled as a multi-label binary classification problem (please see Sec. 2.4). In particular, k input-output bits represent k labels, with each label taking binary 0/1 values, thus there exist 2^k possible classes for the proposed AE framework. Thus, the AE framework aims to design 2^k possible codewords each representing a different class in a higher-dimensional space. For any given hyper-parameter settings, once these codewords are designed the AE framework converges because we can not improve the performance any further. Thus, C_2 is naturally satisfied.

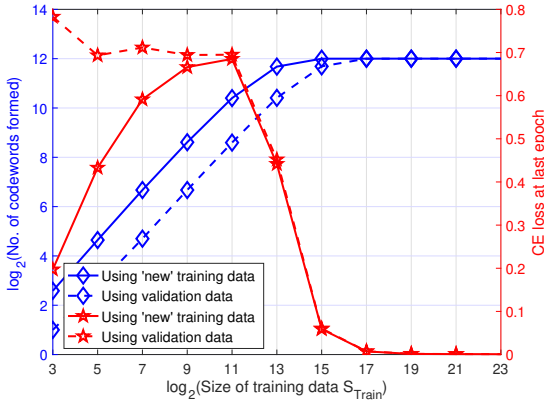
Since we have already shown in Section 2.7 that the proposed AE converges above a minimum required E_b/N_0 . Thus, we focus on showing the relationship between the convergence of the AE frameworks with the training samples below.



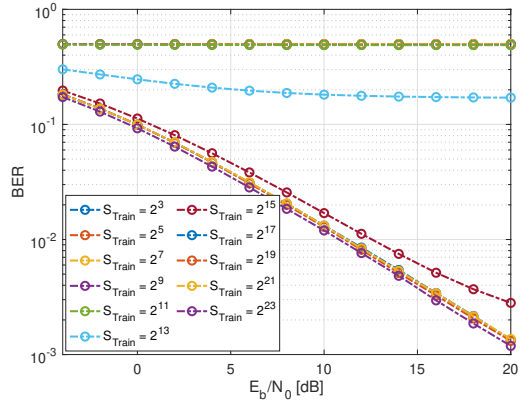
(a) BCM - $(n, k) = (7, 16)$ – Codewords formed by the NN Encoder and binary CE loss on training and validation sets.



(b) BCM - $(n, k) = (7, 16)$ – BER analysis on test set for varying number of Training samples.



(c) d-BCM - $(n, k) = (7, 12)$ – Codewords formed by the NN Encoder and binary CE loss on training and validation sets.



(d) d-BCM - $(n, k) = (7, 12)$ – BER analysis on test set for varying number of Training samples.

Figure 2.5: Proof of the C_1 (second part) and C_2 in Remarks 4 and 5 using the Fixed AE framework for varying rates R .

We now empirically prove the conditions C_1 (second part) and C_2 in Remark 4 below, by considering Fixed AE framework (while similar observations were made for Variable AE framework). For example, we train AE framework performing BCM design with $R = 16/7$ in Fig. 2.5a, 2.5b for varying training data size $S_{\text{Train}} = \{2^{13}, \dots, 2^{23}\}$ of fixed SNR $E_b/N_0 = 25$ dB. Also, we train AE framework performing d-BCM design with $R = 12/7$ in Fig. 2.5c, 2.5d for varying training data size $S_{\text{Train}} = \{2^{13}, \dots, 2^{23}\}$ of fixed SNR $E_b/N_0 = 25$ dB. During the training process, we divide the S_{Train} training samples into 4 : 1 ratio of ‘new’ training set S_T and validation set S_V . Then, we train separate AE frameworks on each of the S_T , and determine the number of codewords formed by the NN encoder and the binary CE loss at the last epoch (during early stopping) on S_T and S_V . Lastly, we determine the BER using the testing samples S_{Test} .

In Fig. 2.5a, 2.5c, we can see that as the training dataset increases, the number of codewords formed by the NN encoder of the trained AE on the training and validation

sets increase until it becomes 2^{16} codewords for $R = 16/7$ (in Fig. 2.5a) and 2^{12} codewords for $R = 12/7$ (in Fig. 2.5c), each representing one of the possible 2^k possible combination, where $k = 16$ for BCM designs and $k = 12$ for d-BCM designs. Moreover, the NN encoder forms these 2^k codewords on the 2^{k+2} and 2^{k+5} training samples using the training and validation sets, respectively. Furthermore, in Fig. 2.5a, 2.5c, we can see that the binary CE loss, noted at the last epoch of training, reduces as the training dataset increases and converges for training and validation sets at 2^{19} training samples.

In Fig. 2.5b, 2.5d, we can see that as the training dataset increases the performance of the proposed AE on the unseen testing samples improves, whereas when the training dataset size starts becoming greater than 2^{k+1} for BCM design (in Fig. 2.5b) and 2^{k+5} for d-BCM design (in Fig. 2.5d), then the performance improvement of the proposed AE starts converging because 2^k codewords are created by the NN encoder of the AE on the training set S_T . The reason the AE framework performing BCM requires $2^4 \times$ less training samples than while performing d-BCM is because – while the d-BCM design the AE also needs to learn to estimate and remove the CSI knowledge effectively.

Thus, for any given hyper-parameter settings, we at least need 2^{k+1} S_{Train} samples for BCM design and 2^{k+5} S_{Train} samples for d-BCM design, to ensure the AE creates 2^k codewords, the validation CE loss has converged, and the AE’s performance converges to its maximum potential of decoding the 2^k possible classes. ■

Remark 7 *As can be seen in the Proof of Remark 4, 5 the number of training samples increases exponentially with input-output bits (k), with at least 2^{k+1} samples required, for the convergence of the AE frameworks with the CSI knowledge. However, these training samples are required only in the offline training phase, where we need to train the AE framework extensively such that it learns the BCM or d-BCM designs to tackle the deteriorating impacts of the fading channel and noise. Once trained, the proposed AE framework can be deployed to predict in online settings.*

2.9 Observations of AE-based BCM and d-BCM Designs

In this section, we perform extensive simulations to analyze the observations of the AE-based BCM and d-BCM designs. The t-stochastic neighbour embedding (t-SNE) [54] is a widely adopted metric in the machine learning (ML) wireless community [12] for insights into the AE-based designed codewords in higher dimensional space, defined as follows:

Definition 15 (t-Stochastic Neighbour Embedding (t-SNE)) *In essence the t-SNE helps us to visualize the $2n$ -dimensional data in 2 dimensions. This happens by transforming the*

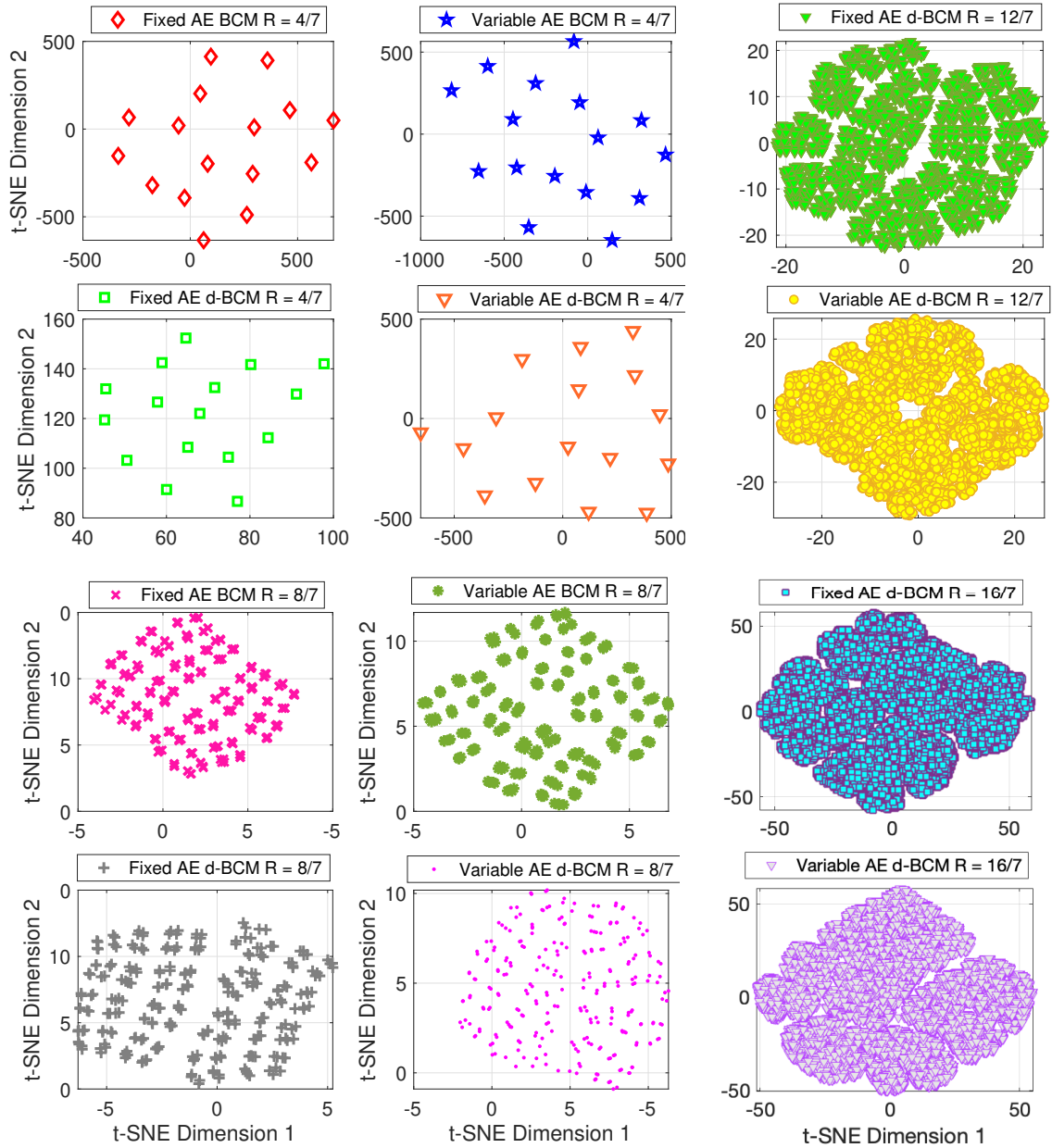


Figure 2.6: t-SNE representation of the codewords formed in the BCM and d-BCM designs using the Fixed and Variable AE frameworks.

similarities among data points to joint probabilities to decrease the KL divergence loss within the joint probabilities of the 2-dimensional embedding and the $2n$ -dimensional codeword design.

For obtaining the AE-based designed codewords, we follow the below procedure. We first train the Fixed and Variable AE frameworks using the NN architecture and hyperparameter settings as described in Sec. 2.6.1, and until convergence using *Remark 4*. Once trained, the NN encoder of the Fixed and Variable AE frameworks becomes deterministic. Thus, if we input any k bits to the NN encoder of the trained Fixed and Variable AE frameworks, we obtain the same n complex baseband symbols (Definition 11) as output every time, representing a codeword (Definition 12) for the k input bits. Now, we can obtain all the possible codewords from the NN encoder of the Fixed and Variable AE frameworks using all the possible combinations of k input bits.

In Fig. 2.6, we plot the t-SNE of the codewords obtained for the BCM and d-BCM designs by the Fixed and Variable AE frameworks for varying rates $R = k/n$. We can see that approximately 2^k clusters are formed in the 2-dimensional space for all the varying rates R using the Fixed and Variable AE frameworks, indicating that approximately 2^k codewords are formed while designing (differential) block-coded modulation for the k bits. Apart from this, we can not obtain any further intuition. Thus, we will focus on the other metrics hereafter.

Since the Fixed AE outperforms the Variable AE in various aspects, in this section, we reveal distinct observations of the Fixed AE-based designed BCM and d-BCM designs. However, a similar analysis can be directly done for the Variable AE framework as well. Throughout this section, we train the proposed Fixed AE framework for various rates $R = k/n$ or combinations of (n, k) , where $n \in \mathcal{N} = \{1, 3, 5, 7, 10\}$ and $k \in \mathcal{K} = \{1, 4, 8, 12, 16\}$, and follow the procedure mentioned above to obtain the Fixed AE-based designed codewords.

In particular, the proposed Fixed AE-based BCM and d-BCM designs exhibit the following distinct observations, for any (n, k) or rate R , as:

Observation – 1: Fixed AE framework designs 2^k codewords in $2n$ -dimensional space.

Proof: In *Proof* of *Remark 4*, we have already shown that the training of the Fixed AE converges after designing of 2^k codewords. Directly, as the NN encoder outputs $2n$ real values for each of the 2^k codewords, i.e. each of the 2^k codewords are represented by unique n complex baseband symbols, thus 2^k codewords are being designed in $2n$ -dimensional space.



Observation – 2: The minimum Euclidean distance between any of the possible codewords, $d_{E_{\min}}$, increases as the rate R decreases. This directly means that

- For any given k bits, the minimum Euclidean distance between any of the possible codewords, $d_{E_{\min}}$, increases as the block length (n) increases.
- For any given block length (n), the minimum Euclidean distance between any of the possible codewords, $d_{E_{\min}}$, increases as the number of bits (k) decreases.

Observation – 3: The minimum Euclidean distance between any of the possible codewords, $d_{E_{\min}}$, is greater for BCM design with the CSI knowledge compared to the d-BCM design without the CSI knowledge.

Observation – 4: For unit block length ($n = 1$), the scenario of only AE-based modulation design, the minimum Euclidean distance between any of the possible codewords, $d_{E_{\min}}$, reduces to zero for the d-BCM designs without the CSI knowledge.

Observation – 5: The AE-based designed BCM and d-BCM reaches its maximum potential of decoding (in terms of BER performance) when $n = \lceil k/2 \rceil$ and $n = \lceil (k + 1)/2 \rceil$, respectively, where $\lceil \cdot \rceil$ denotes ceil function.

Definition 16 (Minimum Euclidean distance ($d_{E_{\min}}$)) *The minimum Euclidean distance between any of the possible 2^k codewords is given as*

$$d_{E_{\min}} = \min_a d_E^a, \quad \forall a \in \{1, \dots, 2^k\} \quad (2.20)$$

where, we determine the minimum Euclidean distance [55] between each $a^{\text{th}} = \{1, \dots, 2^k\}$ codeword and its closest v^{th} codeword, as follows:

$$d_E^a = \min_{v \in \{1, \dots, 2^k\} \text{ and } v \neq a} \|\mathbf{x}_a - \mathbf{x}_v\|_2, \quad \forall a \quad (2.21)$$

where 2^k denotes the number of possible codewords and $\{\mathbf{x}_a, \mathbf{x}_v\}$ denotes the vector comprising n complex values representing the $(\cdot)^{\text{th}}$ codeword of the 2^k possible codewords in the $2n$ -dimensional space.

Proof: For analyzing the *Observations 2–5*, we trained the proposed Fixed AE framework performing BCM and d-BCM designs for ($n \in \mathcal{N}, k \in \mathcal{K}$). Then, using (2.20) we determine the minimum Euclidean distance ($d_{E_{\min}}$) between all the 2^k designed code-

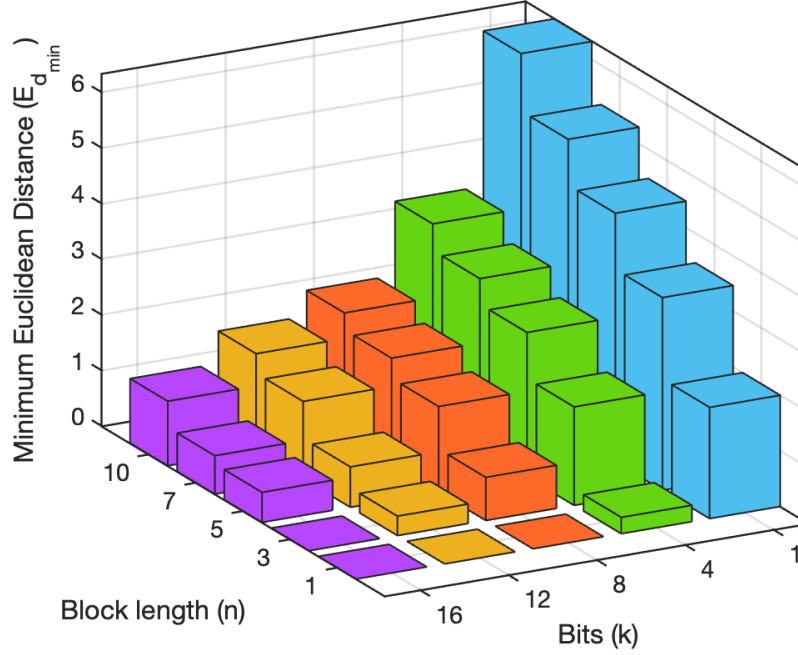


Figure 2.7: Analyzing the minimum Euclidean distance $d_{E_{\min}}$ for varying (n, k) in a BCM design.

words in a BCM design (in Fig. 2.7) and a d-BCM design (in Fig. 2.8). We can see that as the block length (n) increases the $d_{E_{\min}}$ increases and as the number of input bits (k) increases the $d_{E_{\min}}$ decreases, thus the $d_{E_{\min}}$ is inversely proportional to the rate R . This is because the 2^k codewords are being designed in $2n$ -dimensional space.

Further, comparing the minimum Euclidean distance ($d_{E_{\min}}$) for BCM design in Fig. 2.7 and d-BCM design in Fig. 2.8, we can say that for any given rate (R), the $d_{E_{\min}}$ is greater for the BCM design in contrast to the d-BCM design. This is because BCM are designed using the CSI knowledge, while d-BCM are designed without the CSI knowledge. Moreover, in Fig. 2.8 we see that for unit block length ($n = 1$), the $d_{E_{\min}} \approx 0$ for the d-BCM design. This shows that the AE framework is unable to perform differential demodulation with only one symbol, which is very similar to the conventional differential demodulation techniques.

Now, in Fig. 2.9, 2.10, we analyze the BER performance for BCM and d-BCM designs for varying rates $R = k/n$ or combinations of (n, k) . Clearly as the E_b/N_0 increases the BER reduces. Further, in Fig. 2.9 for BCM designs and in Fig. 2.10 for d-BCM designs, we can see the direct impact of increasing minimum Euclidean distance $d_{E_{\min}}$ on the BER performance of the proposed AE framework. Especially, for d-BCM design we can see that for $n = 1$, the AE is unable to decode the signals. This shows that the minimum Euclidean distance $d_{E_{\min}}$ has a direct impact on the BER performance. Moreover, we can see that once the block length (n) becomes sufficiently large there is no BER improvement

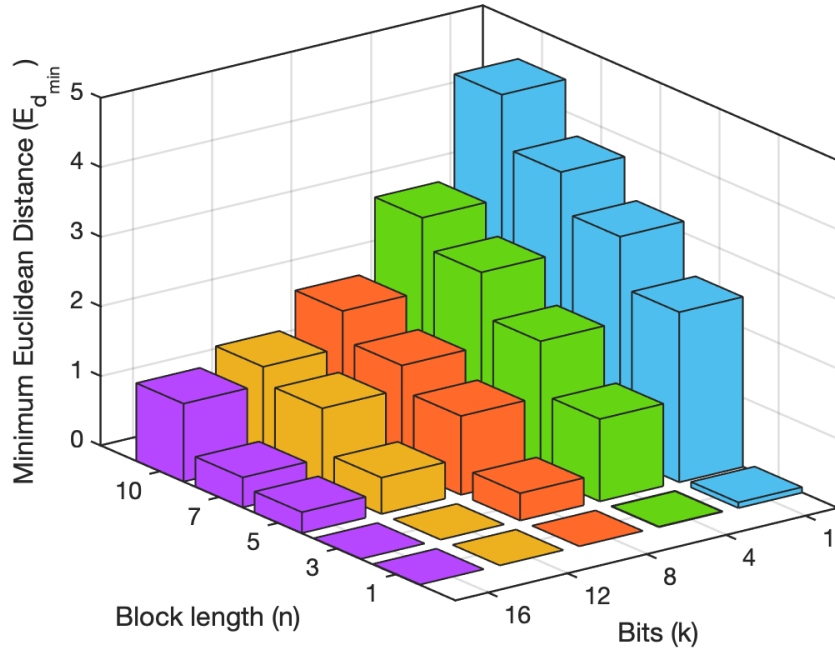


Figure 2.8: Analyzing the minimum Euclidean distance $d_{E_{\min}}$ for varying (n, k) in a d-BCM design.

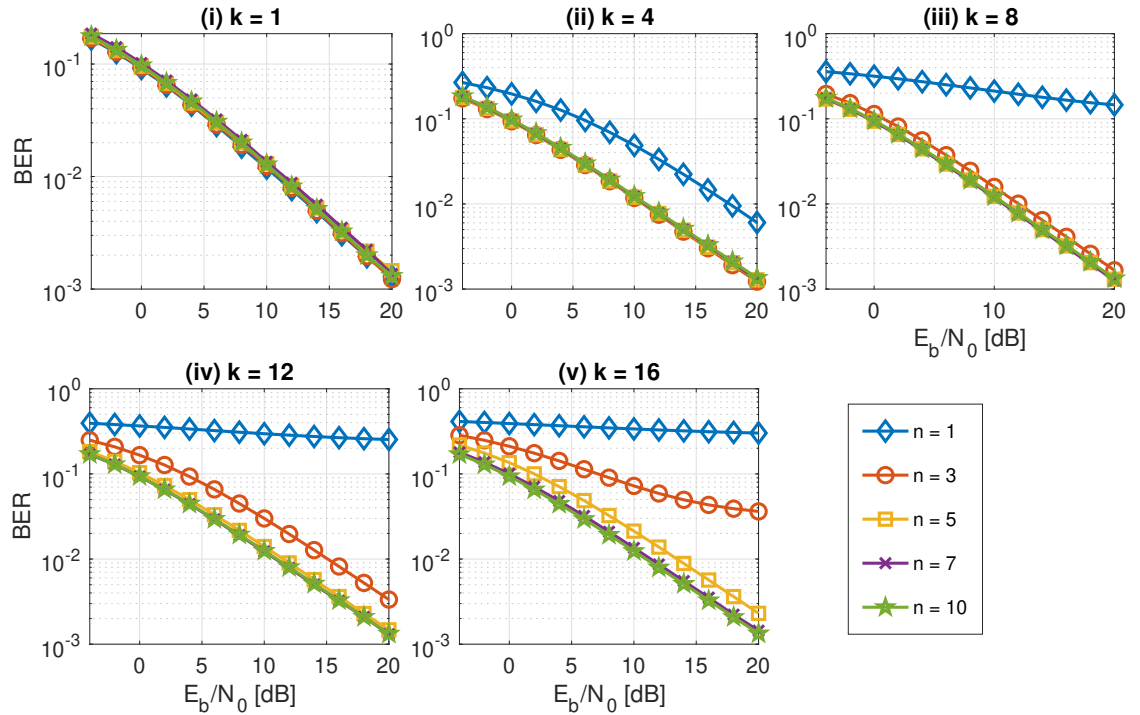


Figure 2.9: Analyzing the BER for varying (n, k) in a BCM design.

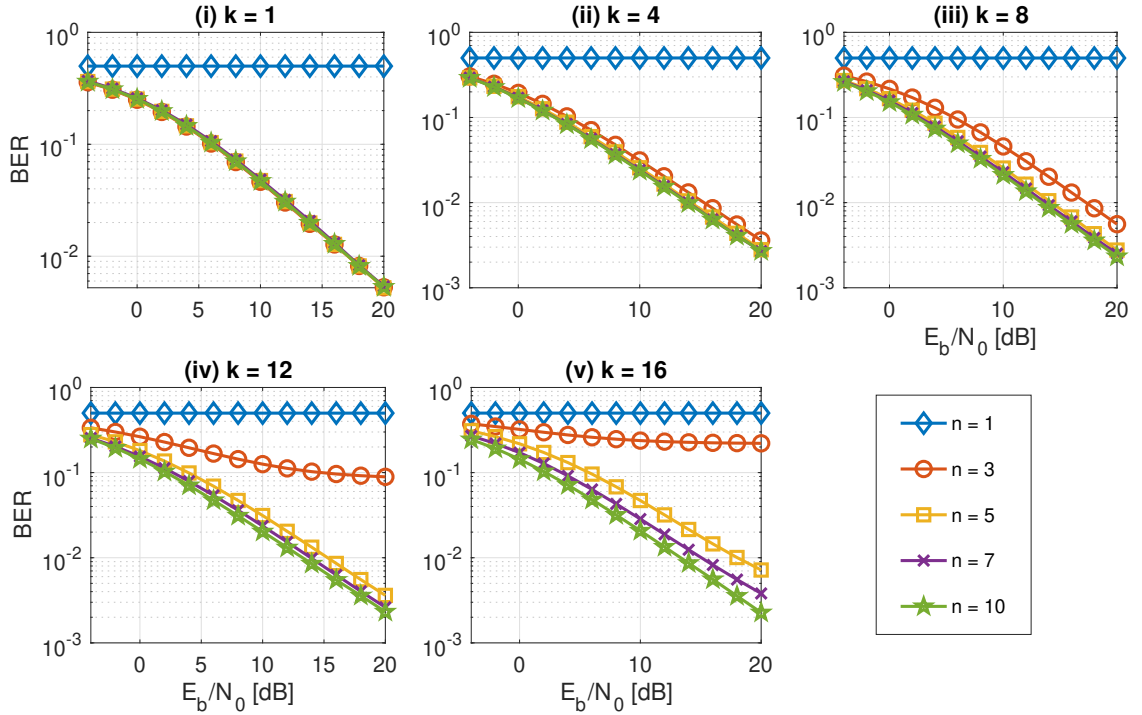


Figure 2.10: Analyzing the BER for varying (n, k) in a d-BCM design.

even with the increasing block length (n). In particular, when the block length becomes half the input bits (k), i.e. $n = \lceil k/2 \rceil$, the BER of the BCM design doesn't improve even with increasing n . Similarly, when the block length becomes $n = \lceil (k+1)/2 \rceil$, the BER of the d-BCM design doesn't improve even with increasing n . This shows that the AE-based designed BCM and d-BCM reaches its maximum potential of decoding (in terms of BER performance) when $n = \lceil k/2 \rceil$ and $n = \lceil (k+1)/2 \rceil$, respectively. This may be because the AE's decoder is unable to decode the signal properly and provide the block length gains.

■

Observation – 6: The minimum Euclidean distance between any of the two closest codewords, d_E^a , varies largely for the d-BCM design than for the BCM design, where $a = \{1, \dots, 2^k\}$.

Observation – 7: When the number of codewords becomes extremely large, the minimum Euclidean distance between any of the two closest codewords, d_E^a , follows a Gaussian distribution for sufficiently large block length (n), where $a = \{1, \dots, 2^k\}$.

Observation – 8: As the block length increases, the Euclidean distance between the any of the two closest codewords, d_E^a , concentrate to the average Euclidean distance, where $a = \{1, \dots, 2^k\}$.

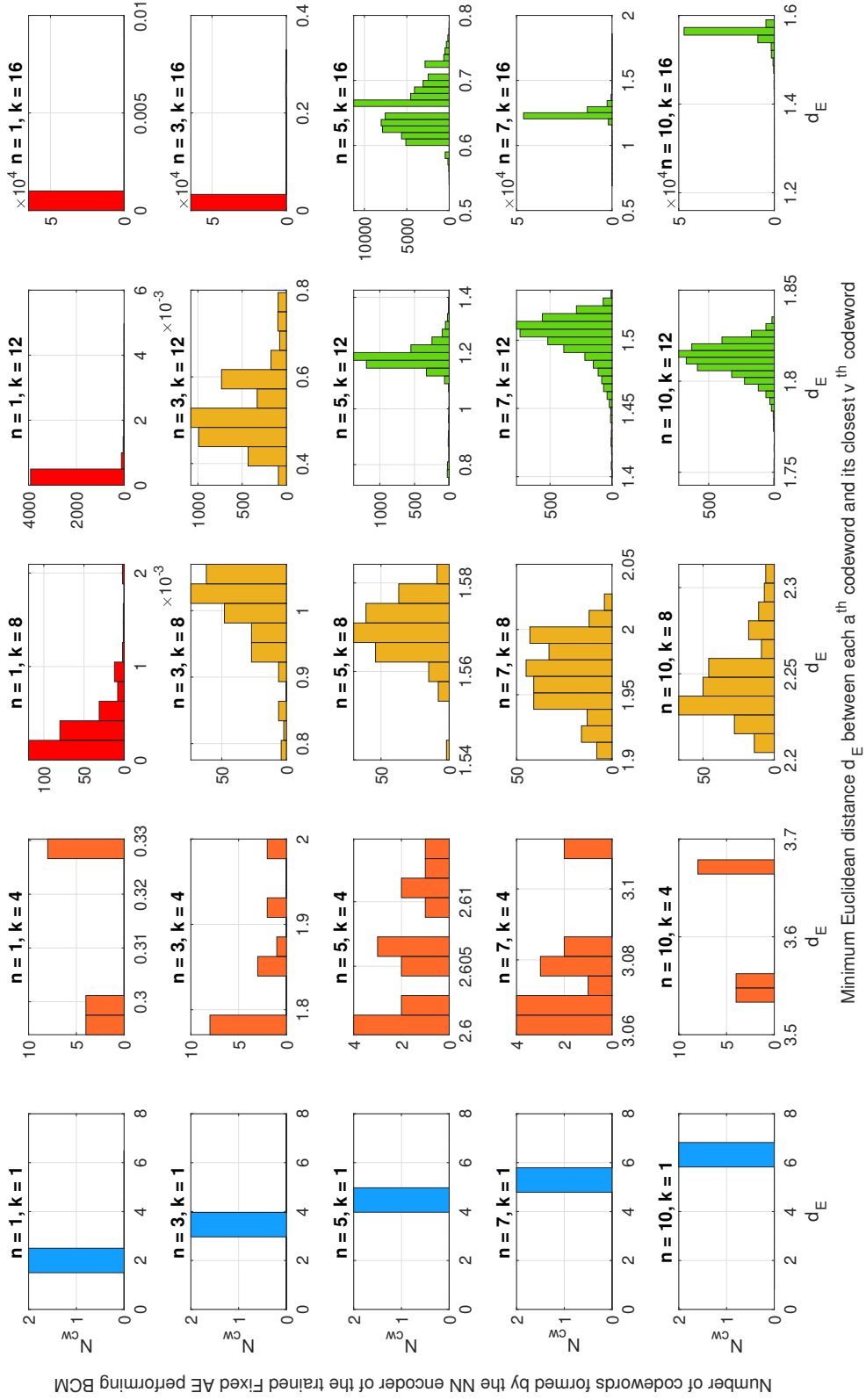
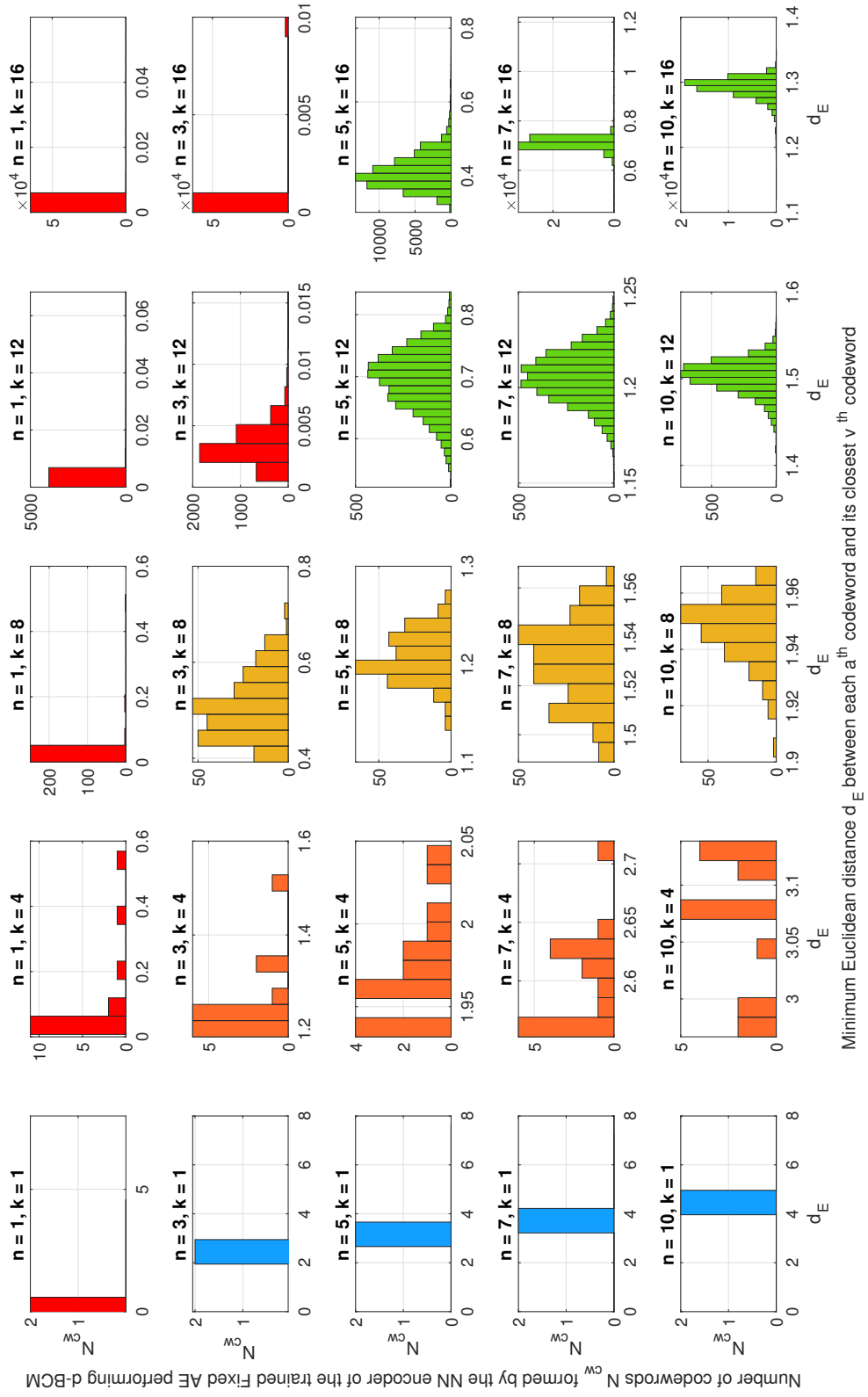


Figure 2.11: Minimum Euclidean distance d_E between each a^{th} codeword and its closest v^{th} codeword in a BCM design.


 Figure 2.12: Minimum Euclidean distance d_E between each a^{th} codeword and its closest v^{th} codeword in a d-BCM design.

Proof: For analyzing the *Observations 6–8*, we trained the proposed Fixed AE framework performing BCM and d-BCM designs for $(n \in \mathcal{N}, k \in \mathcal{K})$. Then, using (2.21) (in *Definition 16*) we determine the minimum Euclidean distance between each a^{th} codeword and its closest v^{th} codeword, to form the $\mathbf{d}_E = \{d_E^1, \dots, d_E^{2^k}\}$ for BCM and d-BCM designs. Lastly, we plot histograms of the \mathbf{d}_E for varying rate R in Fig. 2.11 for BCM design and in Fig. 2.12 for d-BCM design.

Interestingly, the spread of the histogram is greater for the codewords in the d-BCM design without the CSI knowledge (in Fig. 2.12), compared to the codewords in the BCM design with the CSI knowledge (in Fig. 2.11). For example, for $(n = 5, k = 8)$, the histograms spreads from 1.54 to 1.58 for BCM design and from 1.1 to 1.3 for d-BCM design.

Moreover, for the scenarios where number of codewords formed are very large and block length is very small, the $d_{E_{\min}}$ approaches zeros (Fig. 2.7, 2.8) and minimum Euclidean distance between each codeword to its closest codeword is also zero (marked in red in Fig. 2.11, 2.12), thus AE is unable to decode the signal (Fig. 2.9, 2.10). For example, in scenarios like $(n = 1, k = 12)$, $(n = 3, k = 16)$, etc. In such scenarios, the AE learns to *cheat* by placing the 2^k codeword on top of each other because of small space to place the large amount of codewords. The presence or absence of the CSI knowledge also plays a role in determining if the AE will fall for this cheating behaviour, as we can see with $n = 3, k = 12$ in Fig. 2.11, 2.12, the AE is able to design the BCM but is unable to design the d-BCM and thus tries to cheat.

Moreover, in Fig. 2.11 for BCM design and in Fig. 2.12 for d-BCM design, when the block length (n) increases, the mean of histogram also increases because the minimum Euclidean distance $d_{E_{\min}}$ increases. This indicates that as the block length increases, the spacing between any two closest codewords also increases.

Moreover, in Fig. 2.11 for BCM design and in Fig. 2.12 for d-BCM design, when number of codewords becomes extremely high (for $k \geq 8$) and the block length is sufficiently large (for $n \geq 5$), we note the following:

- Although the overall minimum Euclidean distance $d_{E_{\min}}$ (obtained using (2.20)) is small (Fig. 2.7, 2.8), but the minimum Euclidean distance d_E between each a^{th} codeword and its closest v^{th} codeword is competitively large for almost all the codewords and follows a Gaussian distribution (marked in green in Fig. 2.11, 2.12), as a consequence of the central limit theorem (CLT). Specifically, the CLT states that, for sufficiently large sample size, the sampling distribution of the mean for a variable will approximate a normal distribution. This shows that the codewords designed by the NN encoder of the AE follows a normal distribution, making ran-

dom block coded modulations automatically. It is important to note that designing random coded modulations in higher dimensional space using any conventional method was not possible earlier, remaining limited to only 2-dimensional space.

- As the block length increases the standard deviation (spread) of the Gaussian distribution decreases, indicating that the Euclidean distance of the codewords concentrates to the average Euclidean distance, as a consequence of the CLT. As detailed above, the codewords designed by the NN encoder of the AE have minimum Euclidean distance between nearest codewords following a normal distribution. Thus, the CLT states that as the sample size (i.e., the number of codewords) increases, the standard deviation of the sampling distribution (i.e., the minimum Euclidean distance between nearest codewords) becomes smaller because the square root of the sample size (i.e., 2^k) is in the denominator.

These observations resemble the desired theoretical observations of coded-modulation designs discussed in [9]. Therefore, we claim that the proposed AE framework can design random BCM and d-BCM designs to achieve the best possible distance observations for sufficiently large block length (n) for any given input bits (k).

■

Observation – 9: The packing density improves as the rate R decreases.

- For any given k bits, the packing density improves as the block length (n) increases.
- For any given block length (n), the packing density improves as the number of bits (k) decreases.

Observation – 10: The packing density of the BCM design with the CSI knowledge is better than d-BCM design without the CSI knowledge.

Observation – 11: For sufficiently large block length (n), the packing density of the BCM and d-BCM designs converges.

Definition 17 (Normalized second-order moment) We can define the normalized second-order moment (E_n) as the average squared Euclidean distance between a point in the packing and the origin of the coordinate system, normalized by the square of the minimum Euclidean distance, given as

$$E_n = \frac{1}{2^k d_{E_{\min}}^2} \sum_{a=1}^{2^k} \|\mathbf{x}_a\|^2 \quad (2.22)$$

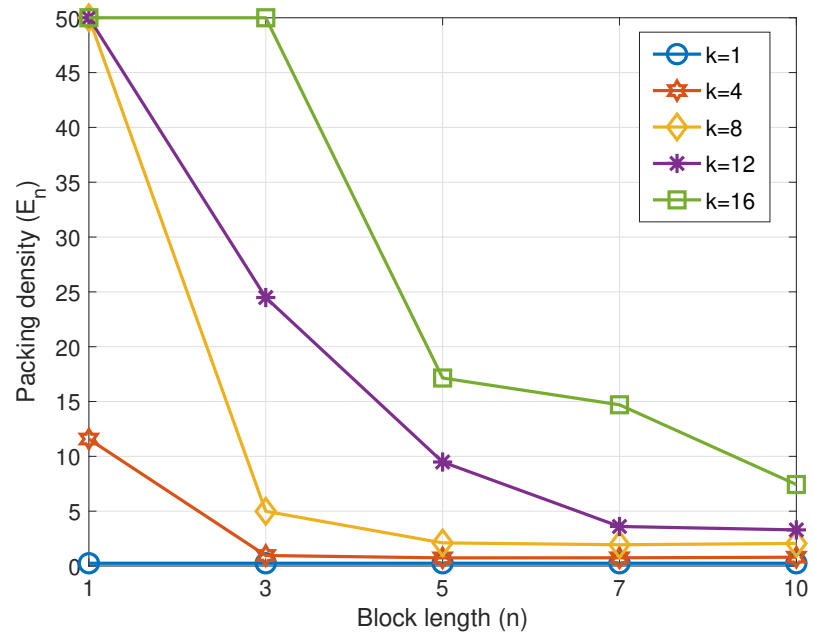


Figure 2.13: Normalized second order moment for varying (n, k) in a BCM design.

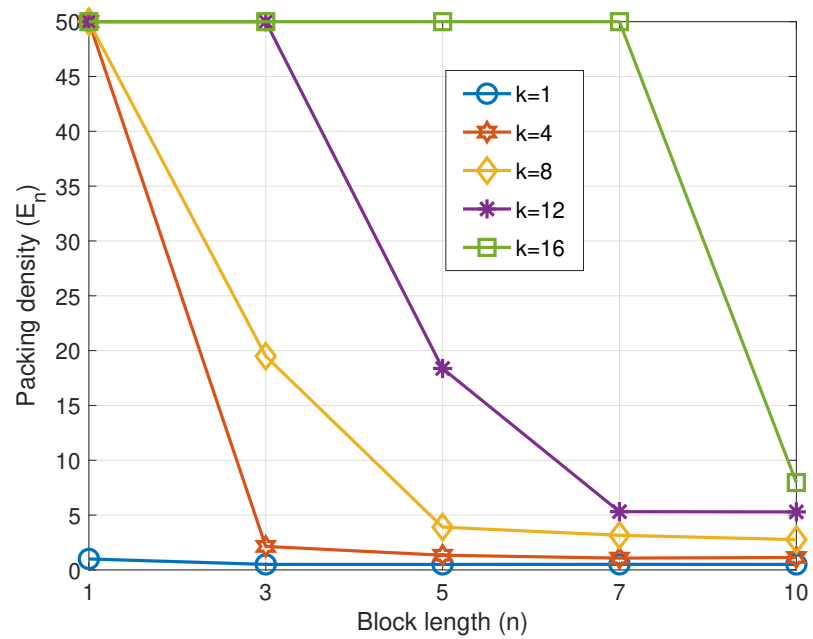


Figure 2.14: Normalized second order moment for varying (n, k) in a d-BCM design.

This metric remains indifferent to scaling thus pivotal to differentiate the packing densities of the designed codewords.

Proof: In Fig. 2.13 and Fig. 2.14 we analyze the packing densities using the normalized second order moment E_n of the BCM design and d-BCM design, respectively, for varying rates $R = k/n$. Please note that lower the E_n the better the packing density. We can see that the packing density improves for the AE-based BCM and d-BCM designs as the block-length (n) increases or the input bits (k) decreases, for all ($n \in \mathcal{N}, k \in \mathcal{K}$), because the 2^k codewords are being designed in the $2n$ -dimensional space. This shows that the packing density improves as the rate R decreases until it starts converging for sufficiently large block length (n).

Also, comparing the packing density of the BCM design in Fig. 2.13 and d-BCM design in Fig. 2.14, we see that, for the same rate R , the packing density of the codewords in the BCM design is better than the d-BCM design, because of the presence of the CSI knowledge in the BCM designs, we almost remove the impact of the fading channels. ■

Observation – 12: The codewords designed by the AE framework are spherical codes.

Definition 18 (Normalized fourth-order moment or Kurtosis) *It measures the variation of the squared Euclidean norm among the codewords, defined as*

$$\chi = \frac{1}{E_n^2 d_{E_{\min}}^4 2^k} \sum_{a=1}^{2^k} \|\mathbf{x}_a\|^4 \quad (2.23)$$

The $\chi = 1$ denotes that a spherical code is created with equal norm for all codewords which is an optimal sphere packing if the number of points per dimension is small enough.

Proof: By simulations, we find that the proposed AE creates ‘Spherical codes’ with $\chi = 1$, i.e. equal norm for all the 2^k codewords for all the varying ($n \in \mathcal{N}, k \in \mathcal{K}$) scenarios, for both the BCM and d-BCM designs. This is because we impose an block power constraint on the NN encoder of the AE using the power normalization layer (\mathbf{P}_N). ■

2.10 Conclusion

In this chapter, we studied AE-based P2P communication networks. We conducted an in-depth literature survey and focussed on notations and definitions of the AE frameworks

that will also be utilized extensively in the rest of this thesis. In particular, we classify the AE frameworks as Fixed and Variable AE if the number of neurons remains independent and dependent on the input-output bits, respectively. Further, we propose both the Fixed and Variable bit-wise AE frameworks under an RBF channel for short block lengths ($n = 7$), performing BCM design with the CSI knowledge and d-BCM design without the CSI knowledge, for high transmission rate R . The training dataset comprises of the RBF channel and multiple transmit SNR values. Also, the end-to-end training is performed by minimizing the binary CE loss, while utilizing step-decay learning rate scheduler and early stopping based on the validation CE loss. We show that the Variable AE is good for low rate $R \leq 8/7$ [bits/channel-reuse], however when the rate becomes high $R > 8/7$ [bits/channel-reuse] then it becomes advantageous to employ the Fixed AE as the number of optimization parameters in the AE reduces, but also for higher rates $R > 12/7$ [bits/channel-reuse] in terms of the BER performance. Also, the BER performance gains of the AE increases with the rate R . Besides, BER performance gains of the AE frameworks without the CSI knowledge is greater than AE frameworks with the CSI knowledge, compared to the conventional networks. Also, Fixed AE has another advantage that it enables us to design a single NN architecture for AE frameworks that can be employed for varying rates R [bits/channel-reuse]. Thus, we will employ a Fixed AE framework in the rest of this thesis.

With a focus on interpretability, we determine the estimated MI of the proposed AE frameworks for varying E_b/N_0 levels. We show that both the AE converges above a minimum required E_b/N_0 depending on the transmission rate and CSI availability. Furthermore, we provide the necessary conditions for AE's training convergence by showing that once the binary cross-entropy validation loss has converged and the NN encoder of AE designs 2^k codewords during the training phase, the AE has converged to its maximum potential of decoding the signal. Lastly, by analyzing the Fixed AE-based BCM and d-BCM designs, we determine distinct observations of the designed codewords in $2n$ -dimensional space, utilizing the metrics such as the minimum Euclidean distance, packing density, and Kurtosis. In the Chapter 3, we will focus on the bit-wise and symbol-wise AE-based BCM and d-BCM designs in a relay network.

Appendix

2.A Autoencoders

Typically, a DL-based Autoencoder (AE) framework comprises of an NN-based encoder and decoder pair, where input data at the NN encoder is aimed to be reconstructed at the NN decoder using the compressed representation, as depicted in Fig. 2.A.1. The NN-based architecture at the encoder is designed such that the input data is compressed to fewer dimensions by removing redundancy from the input data. We also refer to this encoded data as latent space. The NN-based decoder takes the encoded data and aims to reconstruct the input data at the NN encoder. The AE is trained by minimizing the loss between the input data and reconstructed data at the NN encoder and NN decoder, respectively. Since the input data can be either continuous or discrete values, thus the DL-based AE can be designed for regression and classification purposes. Thus, the DL-based AE framework finds applications in compression tasks for varying types of inputs such as for image, video, and speech in Vision community.

2.B Proof of Remark 1

Generally speaking, the optimization problem for the conventional networks (CNs) in (2.10) is considered in two parts, where first we solve the detection problem of 2^{nR} possible codewords that minimizes the symbol-error-rate (SER) and secondly the bit-mapping of 2^{nR} codewords is performed that minimizes the bit-error-rate (BER). In particular, by employing a maximum likelihood detector (MLD) we can obtain the optimum linear block codes that gives us minimum symbol error probability in decoding at the decoder as follows:

$$\begin{aligned}\pi_{\text{ML}}(n, k) &= \min_{\boldsymbol{\theta}_e, \boldsymbol{\theta}_d} \pi^1(n, k) \\ &= \min_{\boldsymbol{\theta}_s, \boldsymbol{\theta}_d} \mathcal{P}[\boldsymbol{\theta}_d(\mathbf{y}_d) \neq v | \boldsymbol{\theta}_s(v), \mathbf{y}_d], \forall v = \{1, \dots, 2^k\}\end{aligned}\tag{2.24}$$

where $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_d$ denotes the encoding and decoding operation rules for MLD. We can solve (2.10) using (2.24) by exhaustively searching all the possible 2^{nR} codewords. Also,

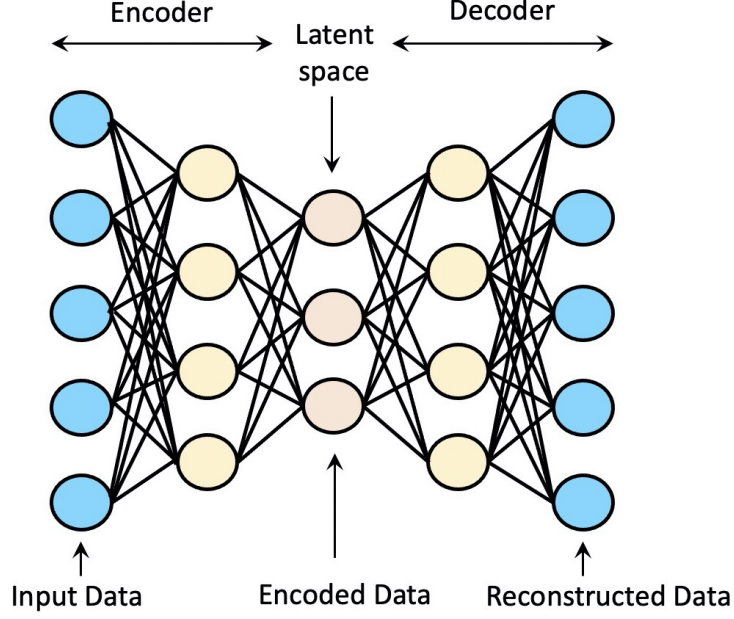


Figure 2.A.1: An illustration of DL-based Autoencoders.

by determining the optimal bit-labelling we can reduce the BER in decoding the $m = \{1, \dots, k\}$ bits as

$$\begin{aligned} \pi_{\text{BL}}(n, k) &= \min_{\boldsymbol{\theta}} \pi^2(n, k) \\ &= \min_{\boldsymbol{\theta}} \mathcal{P}[\boldsymbol{\theta}(\pi^1(n, k)) \neq u_s^m], \forall m = \{1, \dots, k\} \end{aligned} \quad (2.25)$$

where $\boldsymbol{\theta}$ denotes the optimal bit-labelling matrix obtained after solving the $2^k!$ combinatorial problem. Moreover, (2.24) and (2.25) becomes NP-hard to solve for large values of k . If $\boldsymbol{\theta}_s^{\text{CN}_1} = \boldsymbol{\theta}_e$, $\boldsymbol{\theta}_d^{\text{CN}_1} = \boldsymbol{\theta}_d$, and $\boldsymbol{\theta}^{\text{CN}_2} = \boldsymbol{\theta}$ then we can directly say that

$$\pi(n, k) = \pi_{\text{BL}}(\pi_{\text{ML}}(n, k)) \quad (2.26)$$

Now, we only need to prove that minimizing the binary CE loss function of the proposed AE in (2.9) by solving (2.8) gives us same $\boldsymbol{\theta}_e^{\text{CN}_1}$, $\boldsymbol{\theta}_d^{\text{CN}_1}$, $\boldsymbol{\theta}^{\text{CN}_2}$.

Now, let us examine the proposed bit-wise AE framework that takes k bits as input-output as a vector $\mathbf{u}_s \in \{0, 1\}^k$. Where, we utilize a *Sigmoid* activation function $\sigma(x) = \frac{1}{1+e^{-x}}$ on each of the k logits to obtain the probabilities $\tilde{p}_{d_{\boldsymbol{\theta}_d}}(u_s^m | \mathbf{y}_d)$. Using these definitions, we employ the binary CE loss to train the AE as in (2.9). Directly, for the $m^{\text{th}} = \{1, \dots, k\}$ bit (or label) we can say that

$$\mathcal{L}(u_s^m, \tilde{p}_{d_{\boldsymbol{\theta}_d}}(u_s^m | \mathbf{y}_d)) = \begin{cases} -\log(\tilde{p}_{d_{\boldsymbol{\theta}_d}}(u_s^m | \mathbf{y}_d)), & \text{if } u_s^m = 1, \\ -\log(1 - \tilde{p}_{d_{\boldsymbol{\theta}_d}}(u_s^m | \mathbf{y}_d)), & \text{if } u_s^m = 0 \end{cases} \quad (2.27)$$

It is straightforward to say that $\mathcal{L} \left(u_s^m, \tilde{p}_{d\theta_d} (u_s^m | \mathbf{y}_d) \right)$ in (2.27) is minimized when $\mathcal{P} [\boldsymbol{\theta}_d^{\text{AE}}(\mathbf{y}_d) = u_s^m | \boldsymbol{\theta}_s^{\text{AE}}(\mathbf{u}_s), \mathbf{y}_d]$ is maximized, or equivalently $\mathcal{P} [\boldsymbol{\theta}_d^{\text{AE}}(\mathbf{y}_d) \neq u_s^m | \boldsymbol{\theta}_s^{\text{AE}}(\mathbf{u}_s), \mathbf{y}_d]$ is minimized. Thus, we are basically solving the following optimization problem in AE frameworks

$$(\boldsymbol{\theta}_s^{\text{AE}}, \boldsymbol{\theta}_d^{\text{AE}}) = \arg \min_{\boldsymbol{\theta}_s^{\text{AE}}, \boldsymbol{\theta}_d^{\text{AE}}} \mathcal{P} [\boldsymbol{\theta}_d^{\text{AE}}(\mathbf{y}_d) \neq \mathbf{u}_s | \boldsymbol{\theta}_s^{\text{AE}}(\mathbf{u}_s), \mathbf{y}_d] \quad (2.28)$$

Whereas, using (2.26) the optimal conventional networks frameworks' solution can be given as

$$\begin{aligned} (\boldsymbol{\theta}_s^{\text{CN}_1}, \boldsymbol{\theta}_d^{\text{CN}_1}, \boldsymbol{\theta}^{\text{CN}_2}) &= \arg \min_{\boldsymbol{\theta}_s^{\text{CN}_1}, \boldsymbol{\theta}_d^{\text{CN}_1}, \boldsymbol{\theta}^{\text{CN}_2}} \mathcal{P} [\boldsymbol{\theta}^{\text{CN}_2} (\mathcal{P} [\boldsymbol{\theta}_d^{\text{CN}_1}(\mathbf{y}_d) \neq v | \boldsymbol{\theta}_s^{\text{CN}_1}(v), \mathbf{y}_d]) \neq u_s^m], \\ &= \arg \min_{\boldsymbol{\theta}_s^{\text{CN}_1}, \boldsymbol{\theta}_d^{\text{CN}_1}, \boldsymbol{\theta}^{\text{CN}_2}} \mathcal{P} [\boldsymbol{\theta}^{\text{CN}_2} (\boldsymbol{\theta}_d^{\text{CN}_1}(\mathbf{y}_d)) \neq \mathbf{u}_s | \boldsymbol{\theta}_s^{\text{CN}_1}(\mathbf{u}_s), \mathbf{y}_d] \end{aligned} \quad (2.29)$$

Directly, by mapping (2.28) and (2.29), we can say that in the proposed bit-wise AE framework the NN-based encoder is performing coded-modulation design and automatic-bit labelling, whereas the NN-based decoder is performing coded-demodulation for a given NN encoder. Hence, by mapping (2.28) and (2.29), we can say that optimizing $(\boldsymbol{\theta}_s^{\text{AE}}, \boldsymbol{\theta}_d^{\text{AE}})$ in AE-framework is equivalent to designing $(\boldsymbol{\theta}_s^{\text{CN}_1}, \boldsymbol{\theta}_d^{\text{CN}_1}, \boldsymbol{\theta}^{\text{CN}_2})$ for the conventional communication networks. Hence proved.

Chapter 3

AE-based Amplify-and-Forward Relay Networks

3.1 Introduction

Relay networks have appeared as an innovative technology in the past decade owing to their potential of increasing network coverage, reliability, and capacity [17], [18]. The relaying network can be broadly classified as amplify-and-forward (AF) [25], [24] and decode-and-forward (DF) [22], [56] depending on the decoding complexity of the relaying node. The AF relaying is employed practically because it provides low implementation complexity, and the signal is received, amplified, and re-transmitted in the analog domain. Thus in this chapter, we extend the autoencoder (AE)-based point-to-point (P2P) networks (discussed in Chapter 2) by focussing on an end-to-end learning-based AE framework for the AF relay network.

In general, for any rate $R = k/n$ [bits/channel-reuse], the AE frameworks can be broadly classified as the Symbol-wise AE [12] and Bit-wise AE [27], as discussed in Sec. 2.1. The AE-based block coded modulation (BCM) design and differential BCM (d-BCM) design in the presence and absence of the channel state information (CSI) knowledge, respectively, has been investigated using a symbol-wise AE framework for P2P and relay networks only in a handful of works [12, 28, 29, 36]. While in Chapter 2, we introduced the bit-wise AE-based BCM and d-BCM designs even for a P2P network.

Recently, we proposed symbol-wise AE-based AF relay networks by employing NN-based multiple dense layers at the AF relay node [57]. Specifically, we proposed d-BCM designs using a symbol-wise AE framework for $R = 4/7$ [bits/channel-reuse] comparing with BPSK modulation and $(7, 4)$ Hamming code [57]. We show the benefits of AE-based d-BCM over the conventional networks with almost 3 dB BER gains. However, conventionally AF scheme is designed to have lower complexity at the relay node, with just the amplification operation. In contrast, the NN-based processing by using dense

layers at the relay node contradicts the low implementation complexity intended for the AF scheme. Furthermore, we also considered full CSI knowledge at the AF relay node in the AE framework. However, conventionally the AF relay node only has the information about the channel gain knowledge, thereby providing an inherent advantage of utilizing the phase information at the AE-based framework over the conventional AF relay networks. Thus, we summarize the major contributions of this chapter, overcoming the drawbacks of our work [57], below.

- We propose an end-to-end learning-based AF relay network using the AE frameworks. Specifically, we propose to employ NNs consisting of dense layers at the source and destination nodes that constitute the AE framework and employ a conventional AF relay node to minimize the implementation cost and maintain a fair CSI requirement between the proposed and conventional relay networks.
- We design both the symbol-wise AE (detailed in Def. 2) and bit-wise AE (detailed in Def. 3) frameworks for the AF relay assisted network and show that these frameworks are optimized by maximizing the bit-wise mutual information (MI) and symbol-wise MI of an AF relaying network, respectively, while minimizing the relative entropy between the posterior distributions at the encoder and decoder. Later, we formulate the AF relay-assisted AE-based framework as a multilabel (multi-class) classification task for the bit-wise (symbol-wise) AEs.
- For both the AE frameworks, we design BCM and d-BCM designs, depending upon the availability of CSI at the destination node.
- We remove the need for CSI knowledge and noise variances of the links for the proposed AE-based frameworks, even without the NN-based processing at the relay node by performing the d-BCM design in $2n$ -dimensional space. We also propose and investigate the impact of an NN-based radio transformer network (RTN) on the d-BCM design in a dual-hop AF relaying network.
- Further, without CSI knowledge, the amplification factor for the conventional AF relay node becomes a fixed value depending on the second-order statistics of the channel between the source to relay node and noise variances [58, 59, 60, 61, 62, 63, 64]. However, the fixed amplification factor is a sub-optimal approach for the AF scheme. Thus, we also show that by utilizing a power normalization layer (\mathbf{P}_N) (as defined in Definition 7) at the AF relay node, we can improve the process of deciding the amplification factor while keeping the signal transmission reception in the analog domain and removing the requirement of second-order channel statistics and noise variances at the relay and CSI knowledge at the destination node.

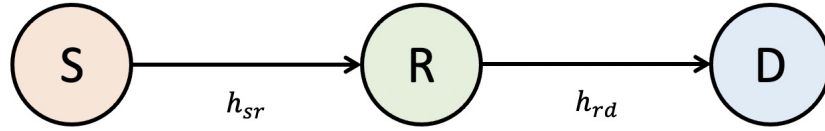


Figure 3.2.1: System model for AF relay networks.

- We focus on interpretability and explain the properties of the $2n$ -dimensional designs by utilizing various metrics like minimum Euclidean distance, normalized second-order and fourth-order moments, and constellation figures of merit. For greater insights, we compare the cross-entropy (CE) loss functions from an information theoretic perspective and the estimated MI to analyze the convergence of the proposed AE frameworks.

3.2 System Model

In this section, we present the conventional AF relay networks, where the source node (S) wants to exchange its intended signal with the destination node (D) by employing an AF relay node (R) in two-phases, as shown in Fig. 3.2.1. Each of the nodes has a single antenna and the direct link between the S and D nodes is strongly attenuated because of severe path-loss and shadowing, and the communication can take place only via the AF relay node (R). We consider block-by-block encoding and decoding operation at the source and destination nodes, whereas the signal transmission in each phase takes place as symbol-by-symbol. The encoder at the source node takes k bits as input and converts it to j bits via channel coding and then transmit the modulated n symbols to the destination node via $2n$ independent channel reuse in two-phases, then the destination node takes n symbols demodulates it to j bits and then perform channel decoding to get the k intended bits. Thus rate for the AF relay network becomes $R = k/2n$ [bits/channel use]. For the sake of clarity in explanation, we consider $n = 1$ in this subsection. We also consider independent and identically distributed (i.i.d.) Rayleigh block fading (RBF) channels $\sim \mathcal{CN}(0, 1)$, such that it remains constant for the n transmissions (block length) in each phase, while changes randomly in each phase and with time.

In the first phase, the source node transmits $\mathbf{u}_s \in \{0, 1\}^k$ bits by mapping \mathbf{u}_s to a complex baseband symbol $x_s = g_m(\mathbf{u}_s) \mapsto \mathbb{C}$, where $g_m(\cdot)$ denotes the modulation process, such that $\mathbb{E}\{|x_s|^2\} = 1$. The signal received by the AF relay node (R) can be given by

$$y_r = \sqrt{P_s} h_{sr} x_s + n_r \quad (3.1)$$

where P_s represents the source transmission power, h_{sr} denotes the channel in the first phase transmission, and n_r is the additive white Gaussian noise (AWGN) at the AF relay node with $n_r \sim \mathcal{CN}(0, \sigma_r^2)$.

In the second phase, the relay node performs symbol-wise amplification with the amplification factor represented as

$$\alpha = (P_s |h_{sr}|^2 + \sigma_r^2)^{-1/2} \quad (3.2)$$

And re-transmit the amplified signal to the destination node (D), given by

$$\begin{aligned} y_d &= \sqrt{P_r} h_{rd} \alpha y_r + n_d \\ &= \underbrace{\sqrt{P_s P_r} h_{rd} h_{sr} \alpha x_s}_{\text{Intended Signal}} + \underbrace{\sqrt{P_s P_r} h_{rd} h_{sr} \alpha n_r + n_d}_{\text{Noise}} \end{aligned} \quad (3.3)$$

where P_r is the transmission power of the relay node, h_{rd} denotes the channel in the second phase transmission, and n_d is the AWGN with $n_d \sim \mathcal{CN}(0, \sigma_d^2)$. The destination node decodes the intended signal \mathbf{u}_s by using the optimal maximum-likelihood detector (MLD), given as follows:

$$\hat{\mathbf{u}}_s = \arg \min_{x \in \mathcal{C}} \left\| y_d - \sqrt{P_s P_r} h_{rd} h_{sr} \alpha x \right\|^2 \quad (3.4)$$

where \mathcal{C} denotes all the possible alphabets, for example $\pm\sqrt{1/2} \pm \sqrt{1/2}i$ (for QPSK), etc.

In the differential scenario, i.e. without the CSI knowledge, we utilize traditional differential modulation and demodulation techniques at the source and destination node, such as differential QPSK (d-QPSK) and MLD decoding. The traditional differential schemes are near optimal because there is no selection combining or cooperative diversity at the destination nodes [58, 59, 60, 61, 62, 63, 64]. However, the amplification factor designed in (3.2) utilizes the channel gain information and noise variances. In the case of absence of the CSI knowledge, there are two distinct ways proposed in literature to design the amplification factor, as detailed below:

- *Transmit power-based amplification factor (TP-based α)* – This approach decides the amplification factor on the basis of only the transmission power of the source node [58, 59], as follows:

$$\alpha = (P_s + 1)^{-1/2} \quad (3.5)$$

- *Channel statistics-based amplification factor (CS-based α)* – This approach utilizes the second-order statistics of the first hop channel between the source and

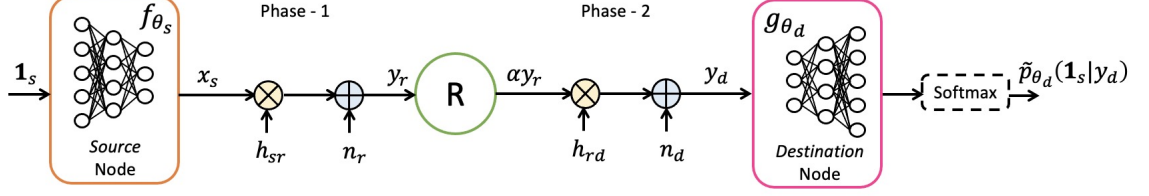


Figure 3.3.1: Symbol-wise autoencoder (SWAE) framework for AF relay networks.

relay node $\sigma_{sr}^2 = \mathbb{E}\{|h_{sr}|^2\}$ and noise variance at the relay node to determine the amplification factor [60, 61, 62, 63, 64], given by

$$\alpha = (P_s \sigma_{sr}^2 + \sigma_r^2)^{-1/2} \quad (3.6)$$

3.3 Proposed AE Frameworks for AF Relay Networks

In this section, we propose the end-to-end learning-based symbol-wise and bit-wise AE frameworks for AF relay network, as detailed in Fig. 3.3.1 and Fig. 3.3.2, respectively.

3.3.1 Symbol-Wise AE (SWAE) Framework for AF Relay Networks

The source node's input message is a one-hot representation vector $\mathbf{1}_s \in \{1, \dots, 2^k\}$ of the 2^k possible symbols of which only one of the element is 1 while the rest are zeros. The source node aims to map the one-hot encoded vector to a complex baseband symbol $x_s \in \mathbb{C}$, by a mapping function $f_{\theta_s}(\mathbf{1}_s, x_s) : \mathbf{1}_s \mapsto x_s$, where f_{θ_s} is the trainable parameters of the NN encoder with weights and bias terms, similar to the Definition 6. We impose a power normalization constraint on the output of the encoder, such that $\|f_{\theta_s}(\mathbf{1}_s, x_s)\|_2^2 = 1$, as detailed in Definition 7. The signal received by the relay node can be given as $y_r = \sqrt{P_s} h_{sr} f_{\theta_s}(\mathbf{1}_s, x_s) + n_r$. The relay node is a conventional AF relay node, thus the received signal is amplified as $x_r = \alpha y_r$. The amplified signal is re-transmitted over the second phase and the signal received by the destination node is given as $y_d = \sqrt{P_r} h_{rd} x_r + n_d$. The destination node implements the de-mapping $g_{\theta_d}(y_d, \tilde{p}_{g_{\theta_d}}(\mathbf{1}_s|y_d)) : y_d \in \mathbb{C} \mapsto \tilde{p}_{g_{\theta_d}}(\mathbf{1}_s|y_d) \in \mathbb{R}^{2^k}$, where g_{θ_d} denotes the trainable parameters of the NN decoder, similar to the Definition 9, and the 2^k outputs represent the decoded one-hot vector representation, also referred as *logits* [41]. These outputs are then passed through a *Softmax* function, $\sigma(\mathbf{x})_{m'} = e^{x_{m'}} / \sum_{m'=1}^{2^k} e^{x_{m'}}$ to obtain the probabilities over the 2^k output message denoted by $\tilde{p}_{g_{\theta_d}}(\mathbf{1}_s|y_d) \in [0, 1]^{2^k}$, such that $\sum_{m'=1}^{2^k} \tilde{p}_{g_{\theta_d}}(\mathbf{1}_s|y_d) = 1$. Now, we utilize the categorical CE loss to train the symbol-wise AE, defined as follows:

Definition 19 (Categorical CE loss [49, 41]) *The categorical CE loss determines the er-*

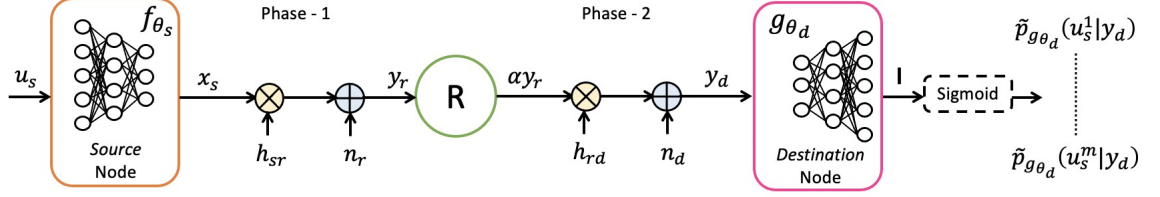


Figure 3.3.2: Bit-wise autoencoder (BWA) framework for AF relay networks.

ror between the input one-hot vector $\mathbf{1}_s$ at the source node and the symbol-wise soft probabilities $\tilde{p}_{g_{\theta_d}}(\mathbf{1}_s|y_d)$, such that, as

$$\begin{aligned} \mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) &:= \mathcal{L}\left(\mathbf{1}_s, \tilde{p}_{g_{\theta_d}}(\mathbf{1}_s|y_d)\right) \\ &= -\sum_{m'=1}^{2^k} \mathbf{1}_s^{m'} \log_2\left(\tilde{p}_{g_{\theta_d}}\left(\mathbf{1}_s^{m'}|y_d\right)\right) \end{aligned} \quad (3.7)$$

Once the symbol-wise AE is trained with the input-output of the network as a one-hot vector $\mathbf{1}_s$ representing the 2^k possible symbol for the k bits. We can obtain the symbol with highest probability as the decoded symbol at the destination node. However, we need to perform bit-labelling separately on the AE-based designed constellation to map the $\hat{\mathbf{1}}_s$ vector to $\hat{\mathbf{u}}_s$ bit vector. But, bit-labelling remains a challenging task, especially as the modulation order increases, or while designing AE-based block coded modulation in $2n$ -dimensional space. Because in such scenarios, the AE-based designed modulation might not form grids as conventional QAM, leading to $2^k!$ possible combinatorial problem to be solved [27].

3.3.2 Bit-Wise AE (BWA) Framework for AF Relay Networks

The source node takes bits as input, given by $\mathbf{u}_s \in \{0, 1\}^k$, and maps it to a symbol $x_s \in \mathbb{C}$, by mapping function $f_{\theta_s}(\mathbf{u}_s, x_s) : \mathbf{u}_s \mapsto x_s$, as detailed in Definition 6. We impose a power normalization constraint on the output of the encoder, such that $\|f_{\theta_s}(\mathbf{u}_s, x_s)\|_2^2 = 1$, as detailed in Definition 7, and the signal received by the relay node can be given as $y_r = \sqrt{P_s}h_{sr}f_{\theta_s}(\mathbf{u}_s, x_s) + n_r$. We consider a conventional relay node, thus the received signal is amplified as $x_r = \alpha y_r$. The amplified signal is re-transmitted over the second phase channel and the signal received by the destination node is given as $y_d = \sqrt{P_r}h_{rd}x_r + n_d$. The destination node implements the demapping $g_{\theta_d}(y_d, \mathbf{1}) : y_d \in \mathbb{C} \mapsto \mathbf{1} \in \mathbb{R}^k$, as detailed in Definition 9. The destination node outputs k logits (one per bit) given by $\mathbf{1} \in \mathbb{R}^k$. Then we apply a *Sigmoid* activation function $\sigma(x) = \frac{1}{1+e^{-x}}$ [41] on each of the k logits, to obtain the probabilities $\tilde{p}_{g_{\theta_d}}(u_s^m|y_d)$, $m = 1, \dots, k$. We train the bit-wise AE by

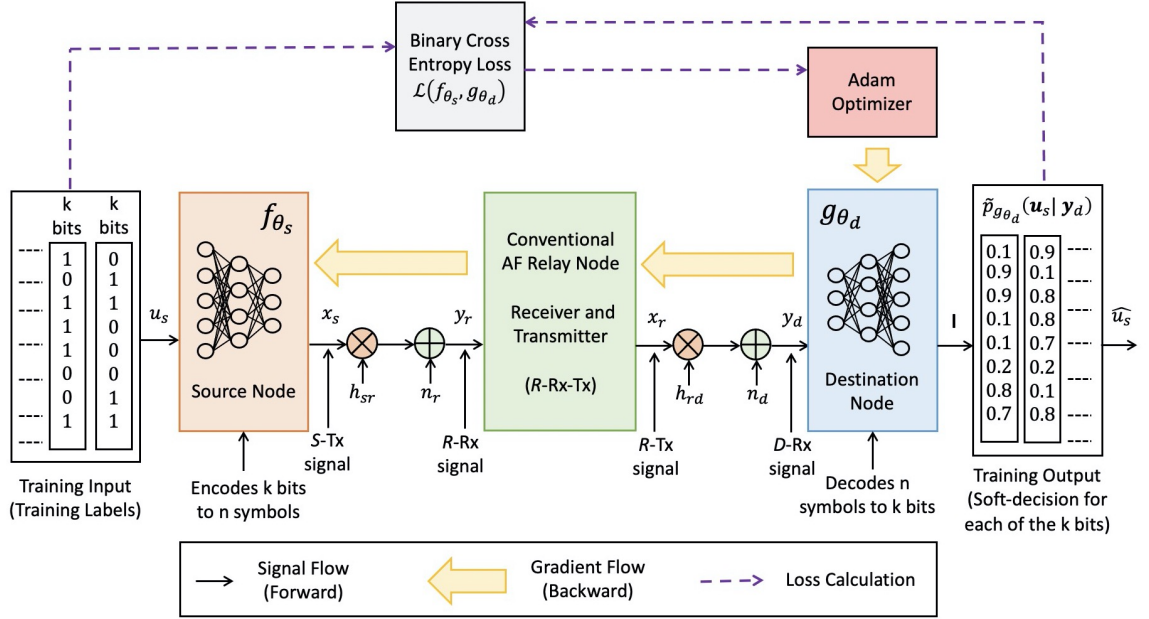


Figure 3.3.3: Block diagram of training for the proposed bit-wise AE-based end-to-end learning system.

minimizing the binary CE loss, as detailed in Definition 13, by keeping $n = 1$.

Once the bit-wise AE is trained with the bits as its input-output, we can directly obtain the bit-labelling for designed complex baseband symbols x_s . We will later show that the bit-wise AE produces gray coded bit-labelling automatically. Thus, bit-wise AE removes the $2^k!$ complexity of bit-labelling required in the symbol-wise AE.

In contrast to the previous bit-wise AE works on P2P networks in [27] where the SNR information was required to train an AE, and two-way AF relay networks in [38] where a separate NN was created for each SNR, we also remove the necessity to acquire the SNR information for correctly encoding or decoding the signals. In fact, we create a single bit-wise or symbol-wise AE framework that can be tested on a range of SNRs reducing the floating parameters in the NN by 95% [57]. Please note we maintain these advantages for designing bit-wise AE in the rest of the thesis as well.

3.4 Implementation of Autoencoder Framework – Parameters and Training

As the output of the bit-wise AE are bits and that of symbol-wise AE are symbols. We can formulate the proposed AF relay assisted AE-based framework as a multilabel (multiclass) classification task for the bit-wise (symbol-wise) AEs. The optimal metric for designing codewords in channel coding is the maximization of the minimum Hamming distance between the codewords. The optimal metric for placing the symbols in the con-

Table 3.4.1: NN architecture of encoder.

Layer No. (l)	Nodes (δ_l^e)	Remarks
$l = 0$	δ_0	Input (e_i)
$l = 1$	64	$\sigma_1 = \text{Tanh}$
$l = 2$	32	$\sigma_2 = \text{Tanh}$
$l = 3$	16	$\sigma_3 = \text{Tanh}$
$l = 4$	$2n$	$\sigma_4 = \text{Linear}$
$l = 5$	$2n$	Power normalization (\mathbf{P}_N)

stellation is the maximization of the minimum Euclidean distance between the symbols. However, we perform joint channel coding and modulation design. Thus, optimizing the bit-wise and symbol-wise AE frameworks by maximizing the minimum Euclidean distance between the constellation points, is not an optimal metric for designing channel-coding [65, 66]. More specifically, optimizing the binary CE and categorical CE losses are better metrics, as they directly optimize the AE frameworks by maximizing the bit-wise MI and symbol-wise MI, respectively [65, 66]. The minimization of the binary CE loss directly leads to the maximization of the generalized MI (as detailed in Chapter 2). It is well known that the maximization of the generalized MI encompasses both the maximization of the minimum Euclidean distance and the maximization of the minimum Hamming distance between the codewords by also including the bit-labeling. Thus, throughout this thesis, we train the AE framework by minimizing the CE loss. Now, we detail the NN architectures for the encoder and decoder, and the process of training the AE frameworks for different CE losses below.

For the sake of brevity, in Fig. 3.3.3 we show the block diagram for the training of the proposed bit-wise AE framework for the AF relay networks, and a similar representation can be done for the symbol-wise AE framework. In this chapter, we utilize L fully-connected (dense) layers, wherein the output of the $l^{\text{th}} \in \{1, \dots, L\}$ dense layer $\omega_l \in \mathbb{R}^{\delta_l}$ can be given as (2.2).

3.4.1 Designing Neural Network-based Encoder

The source node is equipped with its own NN-based encoder that performs block-by-block encoding, wherein k bits (or 2^k symbols) are modulated to n symbols. Since we perform joint channel coding and modulation design, this can be referred as *NN-based block coded modulation design*. The NN encoder consists of $M + 1$ layers with M dense layers and a power normalization (PN) layer, where its NN mapping function $f_{\theta_s}(\mathbf{e}_s, \mathbf{x}_s)$ is represented similar to Definition 6, and given as

$$f_{\theta_s}(\mathbf{e}_s, \mathbf{x}_s) = \mathbf{P}_N(\sigma_M(\mathbf{W}_M \sigma_{M-1}(\dots \sigma_1(\mathbf{W}_1 \mathbf{e}_s + \mathbf{b}_1) \dots) + \mathbf{b}_M)) \quad (3.8)$$

where \mathbf{e}_s represents the input to the encoder and \mathbf{P}_N denotes the power normalization layer, as detailed in Definition 7, with no trainable parameters, mandating that the transmission power is $\|f_{\theta_s}(\mathbf{e}_s, \mathbf{x}_s)\|_2^2 = n$. For the case of symbol-wise AE the input to the encoder becomes symbols, whereas the input to the encoder is bits for the bit-wise AE. Thus, the input to the encoder with the number of input nodes can be given as follows:

$$\{\mathbf{e}_s, \delta_0\} = \begin{cases} \{\mathbf{1}_s, 2^k\}, & \text{for Symbol-wise AE} \\ \{\mathbf{u}_s, k\}, & \text{for Bit-wise AE} \end{cases} \quad (3.9)$$

For the sake of fair comparison between symbol-wise and bit-wise AE frameworks, we keep the encoder's NN architecture same for all the scenarios, as summarized in Table 3.4.1. Note that, although the NN encoder performs block-by-block encoding, the signal transmission takes place symbol-by-symbol.

3.4.2 Designing AF Relay Node

The relay node is designed on the basis of the presence or absence of the CSI knowledge, as follows:

- *With the CSI Knowledge* – In this case, we use a conventional AF relay node that amplifies the received signal using the amplification factor in (3.2) and re-transmits the signal to the destination node. The process of signal transmission–reception remains the same as conventional scenario for both the bit-wise and symbol-wise AEs, as shown in Table 6.2.1. Where, in the first time-slot the source node transmits the signal to the relay node while the relay node remains silent, and in the second time-slot the relay transmits the signal to the destination node while the source node remains silent.
- *Without the CSI Knowledge* – In this case, we propose two approaches as detailed below:
 - *Using Conventional AF Relay* – In this case, we use a conventional AF relay node that amplifies the received signal using the amplification factor in (3.5) or (3.6) depending if the amplification factor is determined using TP-based α or CS-based α method. Further, the process of signal transmission and reception remains the same as conventional scenario for both the bit-wise and symbol-wise AEs, as shown in Table 6.2.1.
 - *Using Power Normalized AF Relay (PN at relay)* – In this case, we modify the signal transmission and reception process, as detailed in Table 3.4.3. Herein, the source node transmits the complete block of data comprising n

Table 3.4.2: Process of signal transmission–reception of a block of data (block length n) for conventional AF relay network, and AE-based scenarios – with CSI knowledge and without CSI knowledge using conventional AF relay.

Time-instants $[\kappa]$	[1]	[2]	[3]	[4]	⋯⋯⋯	$[2n-1]$	$[2n]$
Symbol Tx by S	$x_s[1]$	–	$x_s[2]$	–	⋯⋯⋯	$x_s[n]$	–
Symbol Rx by R	$y_r[1]$	–	$y_r[2]$	–	⋯⋯⋯	$y_r[n]$	–
Symbol Tx by R	–	$x_r[1]$	–	$x_r[2]$	⋯⋯⋯	–	$x_r[n]$
Symbol Rx by D	–	$y_d[1]$	–	$y_d[2]$	⋯⋯⋯	–	$y_d[n]$

 Table 3.4.3: Process of signal transmission–reception of a block of data (block length n) for AE-based scenarios without CSI knowledge using power normalized AF relay.

Time-instants $[\kappa]$	[1]	[2]	⋯⋯⋯	$[n]$	$[n+1]$	$[n+2]$	⋯⋯⋯	$[2n-1]$	$[2n]$
Symbol Tx by S	$x_s[1]$	$x_s[2]$	⋯⋯⋯	$x_s[n]$	–	–	⋯⋯⋯	–	–
Symbol Rx by R	$y_r[1]$	$y_r[2]$	⋯⋯⋯	$y_r[n]$	–	–	⋯⋯⋯	–	–
Symbol Tx by R	–	–	⋯⋯⋯	–	$x_r[1]$	$x_r[2]$	⋯⋯⋯	$x_r[n-1]$	$x_r[n]$
Symbol Rx by D	–	–	⋯⋯⋯	–	$y_d[1]$	$y_d[2]$	⋯⋯⋯	$y_d[n-1]$	$y_d[n]$

symbols to the relay node in the first n time-slots, while the relay node remains silent. Then the relay node performs amplification by normalizing the power of n symbols using the power normalization layer \mathbf{P}_N , as defined in Definition 7, with no trainable parameters, mandating that the transmission power is $\|\mathbf{x}_r\|_2^2 = n$. Then in the next n time slots, the relay node performs symbol-by-symbol transmission, while the source node remains silent.

3.4.3 Designing Neural Network-based Decoder

The symbols re-transmitted by the AF relay node are decoded by the destination node. The decoder performs block-by-block decoding, where n symbols are demodulated to k bits. Since we perform joint channel decoding and demodulation it can also be referred as *NN-based block coded demodulation design*. The decoder at the destination node consists of N dense layers, where its NN de-mapping function $g_{\theta_d}(\mathbf{y}_d, \tilde{p}_{g_{\theta_d}}(\mathbf{e}_s|\mathbf{y}_d))$ is represented similar to Definition 9, and given as

$$g_{\theta_d}(\mathbf{y}_d, \tilde{p}_{g_{\theta_d}}(\mathbf{e}_s|\mathbf{y}_d)) = \sigma_N(\mathbf{W}_N \sigma_{N-1}(\dots \sigma_1(\mathbf{W}_1 \mathbf{L}_L(\mathbf{y}_d) + \mathbf{b}_1) \dots) + \mathbf{b}_N) \quad (3.10)$$

where \mathbf{L}_L denotes the Lambda layers with no trainable parameters, as defined in Definition 10. Please note that the first lambda layer takes received symbols as input and the output of the last lambda layer forms the input to the NN decoder. The Lambda layers changes depending on the presence or absence of the CSI knowledge, thus detailed separately in Section 3.6. For the sake of fair comparison between symbol-wise and bit-wise AE frameworks, we keep the decoder's NN architecture same for all the scenarios, as summarized in Table 3.4.4.

Table 3.4.4: NN architecture of decoder.

Layer No. (l)	Nodes (δ_l)	Remarks
$l = 0$	$2n$	Input (Output of \mathbf{L}_L)
$l = 1$	256	$\sigma_1 = \text{Tanh}$
$l = 2$	128	$\sigma_2 = \text{Tanh}$
$l = 3$	64	$\sigma_3 = \text{Tanh}$
$l = 4$	32	$\sigma_4 = \text{Tanh}$
$l = 5$	δ_0	$\sigma_5 = \text{Softmax (for SWAE) or}$ $\sigma_5 = \text{Sigmoid (for BWAE)}$

3.4.4 Model Training and Updates

The input to the NN encoder at the source node is bits (for BWAE) or symbols (for SWAE), whereas the input to the NN decoder at the destination node is the amplified signal by the AF relay node, distorted in two-phase relay transmission. We train this AE framework, such that it becomes unaffected by testing SNR and can handle the two-hop fading channels effectively. Besides, we are using a conventional AF relay node, thus we do not need to perform explicit training at the AF relay node, unlike the works in [57, 38]. Once trained, we deploy the NN weights at the source and destination nodes for future predictions. Later in this AE-framework, the NN-based source node communicates with the NN-based destination node via an AF relay node.

The expected loss for AE-based AF relay network is given by $\mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d)$. Depending on the constructed symbol-wise or bit-wise AE the loss can be categorical or binary CE loss as shown later in (3.15g) and (3.17), respectively. Similar to the process mentioned in Sec. 2.5.1, we train the AE frameworks to obtain the parameters $\boldsymbol{\theta}_s, \boldsymbol{\theta}_d$ by minimizing the categorical and binary CE losses for BWAE and SWAE frameworks, respectively. Let there be $\{S_{\text{Train}}, S_{\text{Test}}\}$ training and testing samples, then by mini-batch training [41] we can estimate the expected categorical or binary CE loss as

$$\mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) = \frac{1}{B^s} \sum_{q=1}^Q \sum_{b^s=1}^{B^s} \sum_{m=1}^{\delta_0} \mathcal{L} \left(e_s^{(q \times b^s, m)}, \tilde{p}_{d\boldsymbol{\theta}_d} \left(e_s^{(q \times b^s, m)} | \mathbf{y}_d^{(q \times b^s)} \right) \right) \quad (3.11)$$

where B^s denotes the batch size and $Q = S_{\text{train}}/B^s$ represents the number of training samples in a batch. Let us denote the NN optimization parameters as $\Theta = \{\boldsymbol{\theta}_s, \boldsymbol{\theta}_d\}$, then we update the weights using the Adam optimizer as detailed in Sec. 2.5.1.

3.4.5 Predictions

The process of predicting (testing) the S_{Test} testing samples vary depending on the designed bit-wise or symbol-wise AE frameworks as:

- *For Bit-Wise Autoencoders (BWAE)* – We have *Sigmoid* activation at the last layer

of the decoder, giving us soft probabilistic outputs $\tilde{p}_{g_{\theta_d}}(u_s^m|y_d), \forall m = \{1, \dots, k\}$. Thus, the prediction procedure remains the same as Section 2.5.2 to obtain $\hat{u}_s^{(r,m)}$, for each $r^{\text{th}} = \{1, \dots, S_{\text{Test}}\}$ testing sample.

- *For Symbol-Wise Autoencoders (SWAE)* – We have *Softmax* activation at the last layer of the decoder, giving us output $\tilde{p}_{g_{\theta_d}}(\mathbf{1}_s|y_d) \in [0, 1]^{2^k}$. Then for each $o^{\text{th}} = \{1, \dots, S_{\text{Train}}\}$ training sample, we determine the symbol s^o with the largest probability, as follows:

$$s^o = \arg \max_{m'=1, \dots, 2^k} \left(1_s^{(m', o)} \right), \quad \forall o. \quad (3.12)$$

Now, we have to perform bit-labelling, for the 2^k symbols. As detailed earlier bit-labelling can become a $2^k!$ combinatorial problem. Thus, we employ a heuristic approach to label the symbols, wherein we map the symbols to bits $s^q \mapsto \hat{u}_s^q$ according to the Gray codes. For example, for QPSK modulation, i.e. $(n, k) = (1, 2)$, we have 4 symbols as output, we label the symbols as $s_1 \mapsto \{0, 0\}$, $s_2 \mapsto \{0, 1\}$, $s_3 \mapsto \{1, 1\}$, and $s_4 \mapsto \{1, 0\}$, respectively. Then for each r^{th} testing sample we find the symbol s^r with the largest probability, as

$$s^r = \arg \max_{m'=1, \dots, 2^k} \left(1_s^{(m', r)} \right), \quad \forall r. \quad (3.13)$$

And utilize the bit-labelling done for the training dataset to map \hat{u}_s^r from s^r .

Then, we calculate the bit-error-rate (BER) between the true bits intended to be transmitted (\mathbf{u}_s^r) and predicted bits decoded at the receiver ($\hat{\mathbf{u}}_s^r$).

3.5 An Information-Theoretic View of Convergence of the Proposed AE Frameworks

In this section, we demonstrate the impact of varying transmit SNR E_b/N_0 and the convergence of training of the proposed symbol-wise and bit-wise AE frameworks, by giving following Propositions.

Proposition 1 *For a sufficiently large block length (n), the training of the bit-wise and symbol-wise AE frameworks converge to a global minima above a minimum required SNR.*

Proposition 2 *For a sufficiently large block length (n), the training of the bit-wise AE framework converges faster than the symbol-wise AE framework.*

Proof: We follow similar steps as the Proof of Propositions 2–4. For any received block of complex baseband symbols, \mathbf{y}_d , the categorical CE loss in (3.7) for the symbol-wise AE can also be represented as

$$\mathcal{J}^{\text{SWAE}}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) = H\left(p_{s_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|\mathbf{y}_d), \tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|\mathbf{y}_d)\right) \quad (3.14)$$

where $H\left(p_{s_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|\mathbf{y}_d), \tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|\mathbf{y}_d)\right)$ denotes the categorical CE loss between the true distribution $p_{s_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|\mathbf{y}_d)$ at the NN encoder of the source node and the learnt distribution $\tilde{p}_{d_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|\mathbf{y}_d)$ at the NN decoder of the destination node, with parameters $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_d$, respectively. Now, we firstly derive the categorical CE loss for the symbol-wise AE by taking the expectation of (3.14) with respect to \mathbf{y}_d , as follows:

$$\mathcal{J}^{\text{SWAE}}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) := \mathbb{E}_{\mathbf{y}_d} \left[H\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|\mathbf{y}_d), \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|\mathbf{y}_d)\right) \right] \quad (3.15a)$$

$$= -\mathbb{E}_{\mathbf{y}_d} \left[\sum_{m'=1}^{2^k} p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^{m'}|\mathbf{y}_d) \log \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s^{m'}|\mathbf{y}_d) \right] \quad (3.15b)$$

$$= -\sum_{m'=1}^{2^k} \int_{\mathbf{y}_d} p(\mathbf{y}_d) p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^{m'}|\mathbf{y}_d) \log \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s^{m'}|\mathbf{y}_d) d\mathbf{y}_d \quad (3.15c)$$

$$= \sum_{m'=1}^{2^k} \int_{\mathbf{y}_d} p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^{m'}, \mathbf{y}_d) \log \left[\frac{p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^{m'}|\mathbf{y}_d)}{p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^{m'}|\mathbf{y}_d) \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s^{m'}|\mathbf{y}_d)} \right] d\mathbf{y}_d \quad (3.15d)$$

$$= \sum_{m'=1}^{2^k} \int_{\mathbf{y}_d} p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^{m'}, \mathbf{y}_d) \log \left[\frac{p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^{m'}|\mathbf{y}_d)}{\tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s^{m'}|\mathbf{y}_d)} \right] d\mathbf{y}_d - \sum_{m'=1}^{2^k} \int_{\mathbf{y}_d} p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^{m'}, \mathbf{y}_d) \log p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^{m'}|\mathbf{y}_d) d\mathbf{y}_d \quad (3.15e)$$

$$= D_{KL}\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|\mathbf{y}_d) \parallel \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|\mathbf{y}_d)\right) + H_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|\mathbf{y}_d) \quad (3.15f)$$

$$= D_{KL}\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|\mathbf{y}_d) \parallel \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|\mathbf{y}_d)\right) + H(\mathbf{1}_s) - I_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s; \mathbf{y}_d) \quad (3.15g)$$

where $D_{KL}\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|\mathbf{y}_d) \parallel \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|\mathbf{y}_d)\right)$ denotes the Kullback-Leibler (KL) divergence loss between the true distribution $p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|\mathbf{y}_d)$ at the encoder and the learnt distribution $\tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|\mathbf{y}_d)$ at the decoder with parameters $f_{\boldsymbol{\theta}_s}$ and $g_{\boldsymbol{\theta}_d}$, respectively, $H(\mathbf{1}_s)$ denotes entropy of the input symbol-vector $\mathbf{1}_s$ at the source node, and $I_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s; \mathbf{y}_d)$ is the MI between the input symbol-vector $\mathbf{1}_s$ and the received signal at the destination node \mathbf{y}_d with parameters $f_{\boldsymbol{\theta}_s}$.

In (3.15g), the (3.15a) defines the categorical CE, (3.15b) comes from the definition of CE, (3.15c) opens the expectation along \mathbf{y}_d , in (3.15d) we multiply and divide by $p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^{m'}|\mathbf{y}_d)$, in (3.15e) we open the log function, (3.15f) comes from the definition of KL divergence and entropy, and (3.15g) utilizes the identity $H(a|b) = H(a) - I(a; b)$ [49].

By minimizing the categorical CE loss in (3.15g), we are optimizing the symbol-wise AE framework to maximize the symbol-wise MI between input symbol vector at the source node and received signal at the destination node $I_{f_{\theta_s}}(\mathbf{1}_s; \mathbf{y}_d)$, while minimizing the KL-divergence $D_{KL}\left(p_{f_{\theta_s}}(\mathbf{1}_s|\mathbf{y}_d)||\tilde{p}_{g_{\theta_d}}(\mathbf{1}_s|\mathbf{y}_d)\right)$ between the posterior distributions learned at the encoder and the prior distribution learned at the decoder. Further, the entropy of the one-hot vector $H(\mathbf{1}_s)$ remains constant.

We can now obtain the estimated MI as detailed in Definition 14. Please note that for the AF relay network, the estimated MI will be divided by 2 because the transmission takes place in two time-slots or two-phases. Thus estimated MI for symbol-wise AE becomes $\mathcal{I}^{\text{SWAE}} = \tilde{I}^{\text{SWAE}}(\mathbf{1}_s; \mathbf{y}_d)/2$. By rearranging the terms in (3.15g) we have

$$\begin{aligned}\tilde{I}_{f_{\theta_s}}^{\text{SWAE}}(\mathbf{1}_s; \mathbf{y}_d) &\triangleq I_{f_{\theta_s}}(\mathbf{1}_s; \mathbf{y}_d) - D_{KL}\left(p_{f_{\theta_s}}(\mathbf{1}_s; \mathbf{y}_d)||\tilde{p}_{g_{\theta_d}}(\mathbf{1}_s; \mathbf{y}_d)\right) \\ &= H(\mathbf{1}_s) - \mathcal{J}^{\text{SWAE}}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d)\end{aligned}\quad (3.16)$$

Since the first term in (3.16) remains a constant, we can see that the changes in the estimated MI of a symbol-wise AE depends on the changes in the categorical CE loss only.

In the bit-wise AE framework, we minimize the binary CE loss, thus we can directly utilize the Proof of Propositions 2–4 here. For the sake of brevity, herein we show the final results. We firstly derive the binary CE loss for the bit-wise AE by taking the expectation of (2.16) with respect to \mathbf{y}_d , as follows:

$$\begin{aligned}\mathcal{J}^{\text{BWAE}}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) &:= \sum_{m=1}^k \mathbb{E}_{\mathbf{y}_d} \left[H\left(p_{f_{\theta_s}}(u_s^m|\mathbf{y}_d), \tilde{p}_{g_{\theta_d}}(u_s^m|\mathbf{y}_d)\right) \right] \\ &\triangleq D_{KL}\left(p_{f_{\theta_s}}(\mathbf{u}_s|\mathbf{y}_d)||\tilde{p}_{g_{\theta_d}}(\mathbf{u}_s|\mathbf{y}_d)\right) + H(\mathbf{u}_s) - I_{f_{\theta_s}}(\mathbf{u}_s; \mathbf{y}_d)\end{aligned}\quad (3.17)$$

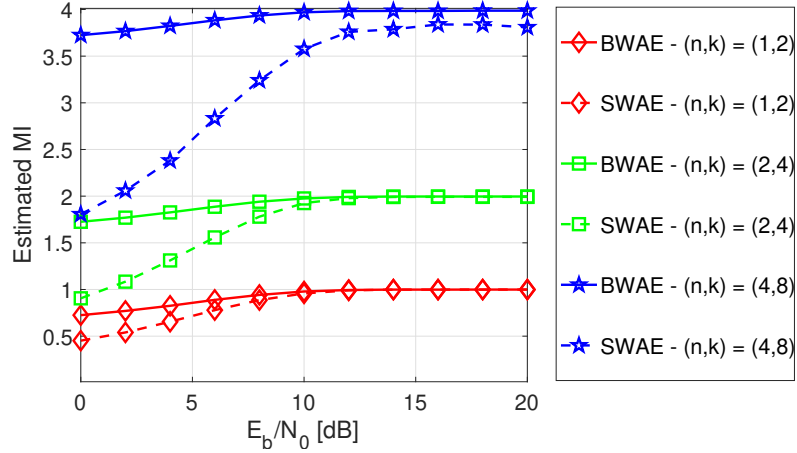
Next, we determine the estimated MI for the bit-wise AE for the AF relay network, given as $\mathcal{I}^{\text{BWAE}} = \tilde{I}^{\text{BWAE}}(\mathbf{1}_s; \mathbf{Y}_d)/2$, divided by 2 because of dual-hop transmission, where

$$\tilde{I}^{\text{BWAE}}(\mathbf{1}_s; \mathbf{y}_d) = H(\mathbf{u}_s) - \mathcal{J}^{\text{BWAE}}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d)\quad (3.18)$$

Similar to the symbol-wise AE's estimated MI, the changes in the estimated MI of a bit-wise AE depends on the changes in the binary CE loss only.

Lastly, by simulations, we analyze the convergence of the training of proposed symbol-wise and bit-wise AE frameworks. In particular, we train a separate symbol-wise (or bit-wise) AE for each E_b/N_0 (in dB) level. Once the symbol-wise and bit-wise AEs are trained we note the validation CE loss $\mathcal{J}^{\text{SWAE}}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d)$ and $\mathcal{J}^{\text{BWAE}}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d)$ at the last epoch of the training, and obtain the estimated MIs $\mathcal{I}^{\text{SWAE}}$ and $\mathcal{I}^{\text{BWAE}}$, respectively.

We now compare the estimated MI for the symbol-wise and bit-wise AE for the AF relay networks in Fig. 3.5.1. We keep $(n, k) = \{(1, 2), (2, 4), (4, 8)\}$ under AWGN chan-


 Figure 3.5.1: Estimated MI versus transmit SNR E_b/N_0 .

nels. As the transmit SNR increases the estimated MI increases, until it reaches the upper bound $k/2$. This suggests the KL-divergence loss approaches 0 as E_b/N_0 increases indicating that the proposed AE frameworks can well approximate at a moderate E_b/N_0 , but bit-wise AE shows ≈ 6 dB improvement in estimated MI at $E_b/N_0 = 0$ dB for AE-based modulation design ($n = 1$). The improvement of bit-wise AE further increased with AE-based block modulation design ($n > 1$) where we see that bit-wise AE shows ≈ 10 dB improvement in estimated MI at $E_b/N_0 = 0$ dB. As the entropy term in (3.16) and (3.18) is equal, the estimated MI depends only on the classification errors (or the CE losses) calculated across the 2^k input-output for the symbol-wise AE compared to k input-output in the bit-wise AE, and the low SNR regime leads to more classification errors in AE training with the additional $(2^k - k)$ number of classes in a symbol-wise AE.

Thus, minimizing the binary CE loss for the bit-wise AE leads to faster convergence of the AE to global minima compared to the minimization of the categorical CE loss for the symbol-wise AE, while both the AEs converge to a global minima at a moderate SNR.

3.6 Numerical Results

In this section, we evaluate the proposed bit-wise and symbol-wise AE frameworks for the AF relay networks with practical SNR values. We utilize QPSK modulation similar to [36]. To train the proposed architectures we utilize SGD with Adam optimizer [42], where the weights of the dense layers are initialized with the Glorot initializer [45]. We keep the learning rate $\tau = 0.00125$, batch size $B = 6000$, number of training epochs as 50, and transmission power of each node $P_s = P_r = 1$. We create the training and testing dataset, comprising of transmitted bits, fading channels, and noise with different E_b/N_0 , in the MATLAB. While we implement the proposed AE framework using Keras [43] with Tensorflow [44] as back-end. We train the AE framework using the training dataset

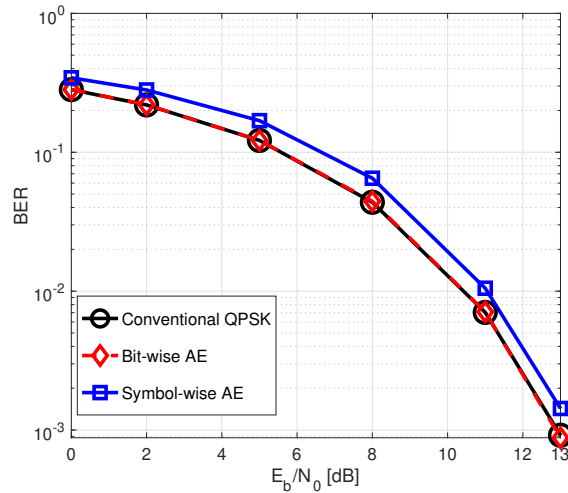
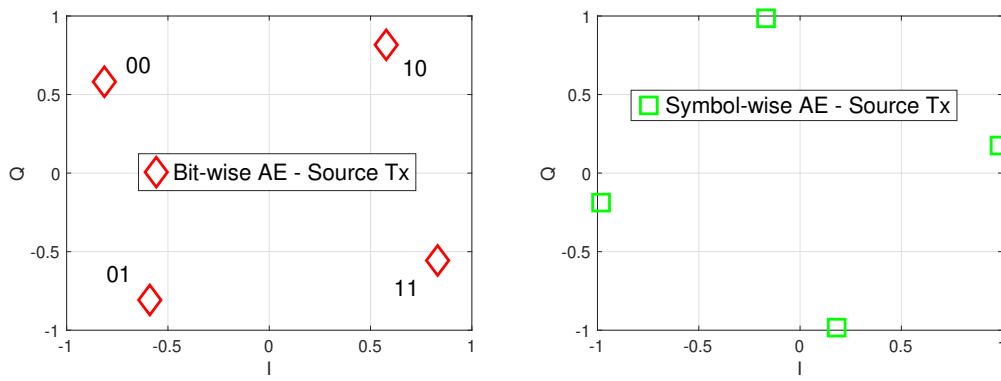
created in MATLAB. We show the performance for AWGN and RBF channels, where the channel remains constant during a transmission block of n symbols and then changes randomly. For the conventional scenarios, we utilize the optimal MLD in (6.5) at the destination node.

We have created the bit-wise and symbol-wise AE for the AF relay networks such that it remains *unaffected of testing SNR values*. In other words, we create a single AE model that can be deployed for any testing SNR. Thus, unlike the previous works in [27, 38], our proposed symbol-wise and bit-wise AE frameworks do not need the SNR value for prediction. For this, we create a training dataset of 10^5 samples for each of the SNR values in \mathcal{S} , and we test the designed models on the unseen testing dataset of 10^5 samples. For AWGN channels, we keep the $\mathcal{S} = \{4\}$ dB and for the RBF channels, we keep the $\mathcal{S} = \{3, 10, 23, 28, 38, 42\}$ dB. We note that for AWGN channels, only one SNR point brings the best AE performance, in particular, a low SNR point, intuitively this is because in the presence of large AWGN the AE learns to map the constellation points as far away as possible. For the RBF channels, we need multiple SNR points to obtain the best AE performance, ranging from low SNR to high SNR points, intuitively this is because the AE needs to learn to map constellation in the presence of both RBF channels in two-hop transmission and AWGN at the relay and destination node, thus when training only a low SNR point the AE observes only noise and thereby is not able to learn to map the constellation optimally leading to the stagnation in BER curves, especially in differential scenarios.

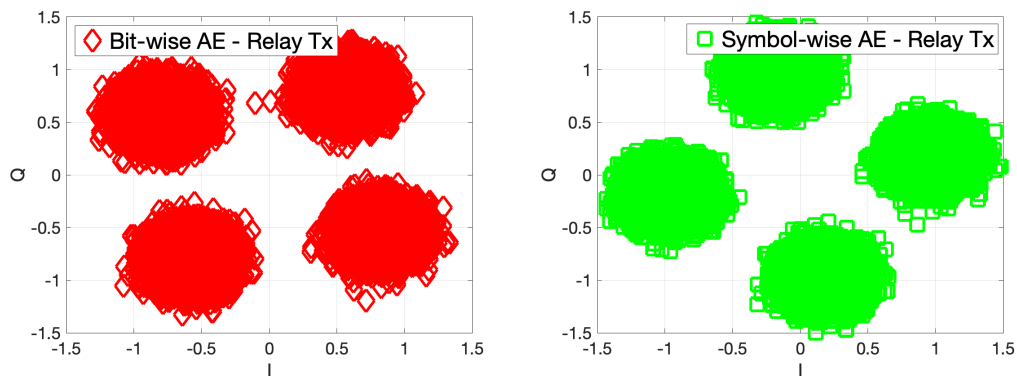
3.6.1 AE-based Modulation Design of the AF relay Network under AWGN channels

In this subsection, we evaluate the proposed *AE-based modulation design*, i.e. we keep the number of channels reused $n = 1$. Particularly we keep $(n, k) = (1, 2)$ and for sake of clarity, we utilize AWGN channels by considering $h_{(\cdot)} = 1$, this assumption holds as there is no direct link between the source and destination link. Also, there are *no* Lambda layer at the decoders.

We show the transmit SNR (E_b/N_0) versus the BER performance in Fig. 3.6.1a. As the SNR increases the BER reduces. As we know that MLD is optimal for AWGN channels, we can see that the proposed bit-wise AE achieves performance similar to the optimal MLD of the conventional AF relay networks. Whereas the proposed symbol-wise AE performs ≈ 1 dB worse than the optimal MLD. This can be understood by the constellation learnt by the AEs in Fig. 3.6.1b. The constellation learned by the encoder of the source node in both the proposed bit-wise and symbol-wise AEs are four symbols for four possible combinations of bits ($k = 2$). Thus, the NN-based encoder forms 2^k symbols for


 (a) BER versus E_b/N_0 [dB].


(b) Signal transmitted by source (S) node.



(c) Signal transmitted by relay (R) node.

 Figure 3.6.1: Performance evaluation for the AE-based modulation design, with $(n, k) = (1, 2)$, for the AF relay networks.

k input bits in both the bit-wise and symbol-wise AE. Further, bit-wise AE is leading to an automatic bit-labeling in Gray coding format, whereas bit-labeling for symbol-wise AE is done heuristically as detailed in Section 3.4.5. Furthermore, bit-wise AE leads to optimal rotation and translation leading to the performance gains, compared to symbol-wise AE.

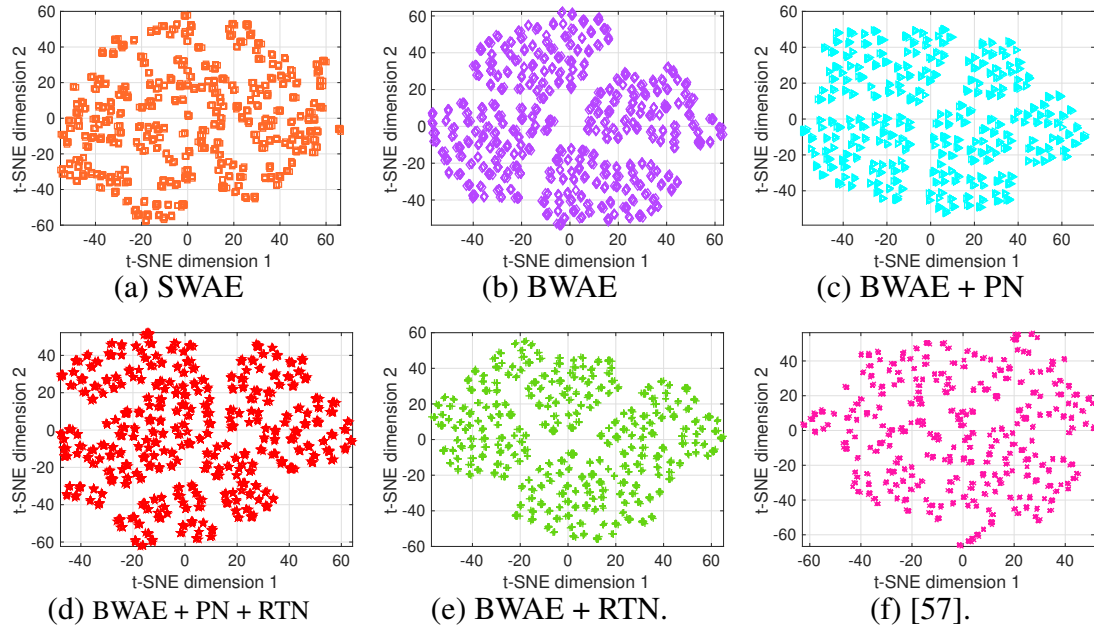


Figure 3.6.2: t-SNE representation in 2 dimensions.

3.6.2 AE-based d-BCM Design under RBF channels

In this subsection, we evaluate the performance of the proposed bit-wise and symbol-wise AE-based d-BCM design, i.e. we keep the number of channels reused $n > 1$, in particular, we keep $(n, k) = (7, 8)$, under RBF channels. Herein we consider a differential scenario, thus none of the (S, R, D) nodes has the CSI knowledge or noise variance information for any links. We perform AE-based d-BCM designs. For the conventional scenarios, we consider (7, 4) Hamming coding, along with differential QPSK and MLD decoding. We utilize TP-based α in (3.5) and CS-based α in (3.6) for the conventional and the proposed AE frameworks. Note that we do not have any Lambda layers at the decoder.

The *t-SNE* [54] is a widely adopted metric in the ML wireless community [12], [57], [67] for insights into the AE-based designed constellations in higher dimensional space, defined as *Definition 15*. From Fig. 3.6.2, we can see that approximately $256 = 2^8$ clusters are formed in the 2-dimensional space for all the proposed AE models, indicating that approximately 2^k codewords are formed while designing block coded modulation for k bits in bit-wise AE and 2^k symbols in symbol-wise AE frameworks. Apart from this, we can not obtain any further intuition (similar to Chapter 2). Thus, we will focus on the other metrics hereafter.

For greater insights, we propose to evaluate the following metrics generally used for sphere packing¹ to characterize the optimality of designed (differential) block coded-

¹We can define the problem of sphere packing as the packing of M points in a $2n$ -dimensional space, with each point considered as the center of a $2n$ -dimensional hypersphere of a given diameter and the aim remains to pack each of these spheres as densely as possible without overlapping, such that the Euclidean distance between any two points is above a defined value. Generally speaking, the problem of sphere packing increases with increase in n dimensions and becomes infeasible for large values of n . Whereas, the

modulation in $2n$ -dimensions [55]: minimum Euclidean distance $d_{E_{\min}}$ (see *Definition 16*), normalized second order moment E_n (see *Definition 17*), normalized fourth order moment or Kurtosis χ (see *Definition 18*), and

Definition 20 (Constellation figure of merit) *This metric is the most suitable energy metric as the modulation constructions are being analyzed at the identical bandwidth, given as*

$$CFM = n/E_n \quad (3.19)$$

Conventional versus Proposed AE Frameworks

We compare the performance of conventional (d-QPSK + (7, 4) Hamming coded) and proposed AE frameworks using TP-based α in (3.5), in Fig. 3.6.3². From Table 3.6.1, we can see that the minimum Euclidean distance for the designed symbol-wise AE is 1.30 compared to only 0.76 in the conventional scenario, still the symbol-wise AE performs ≈ 0.8 dB worse than the conventional scenario. This can be explained by *Remark 8* below.

Remark 8 *From Table 3.6.1 we have $\chi = 1$ indicating that a symbol-wise AE performs d-BCM design as spherical codes in $2n = 14$ -dimensional space. Which is not of the form of a grid, thereby leading to a $2^8!$ combinatorial-problem for bit-labeling the designed codewords in the $2n$ -dimensional space. Although we utilize a heuristic method to label the bits in codewords as the gray-coding format, it is still sub-optimal leading to the performance degradation of the symbol-wise AE.*

In Fig. 3.6.3, we can see that the bit-wise AE performs ≈ 3.5 dB better than the conventional scenario. This can be explained by the following *Remark 9*.

Remark 9 *Firstly, we utilize the minimization of the binary CE loss which is equivalent to maximizing the generalized mutual information (GMI) of the relay network (see Chapter 2). Secondly, we have shown in (3.17) that by minimizing the binary CE loss we maximize the bit-wise MI, which is closely related to achievable rate by the bit-metric decoding (BMD) [27]. Thirdly, we are performing d-BCM design, thus the constellation design is taking place in $2n$ -dimensional space, which from the modulation perspective, is leading to the maximization of the minimum Euclidean distance and minimization of packing density of the points to 1.32 and 3.9 in bit-wise compared to the 0.76 and 12.1 in*

AE-based methods provide us an easy solution to learn the mappings in higher dimensional space.

²Please note that we see similar performance gains by using CS-based α in (3.6), thus for the sake of brevity, we show the performance with only TP-based α in Fig. 3.6.3.

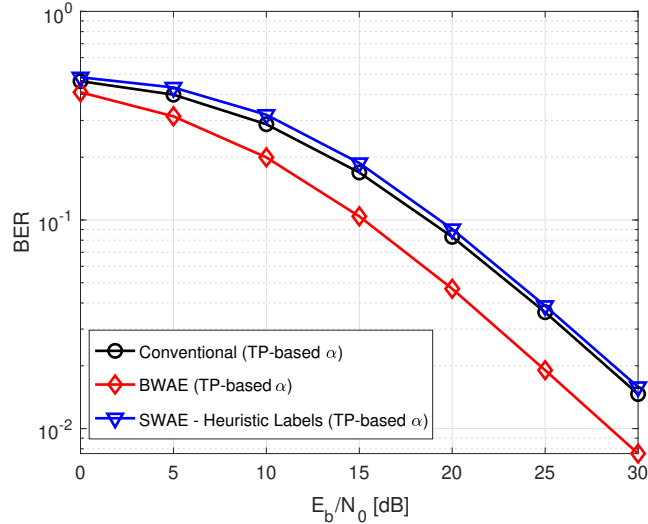


Figure 3.6.3: Conventional versus Proposed AE Frameworks.

Table 3.6.1: Insights using other metrics for d-BCM design

Model	Minimum Euclidean Distance $d_{E,\min}$ (\uparrow better)	Normalized second moment E_n (\downarrow better)	Normalized fourth moment χ	Constellation Figure of Merit CFM (\uparrow better)
Conventional Diff. QPSK + Hamming Code	0.76	12.1	1	0.6
State-of-the-art [57]	0.9	8.6	1	0.8
Symbol-wise AE	1.30	4.1	1	1.7
Bit-wise AE	1.32	3.9	1	1.8
Bit-wise AE + PN at relay	1.30	4.1	1	1.7
Bit-wise AE + RTN at dest.	1.27	4.3	1	1.6
Bit-wise AE + PN at relay + RTN at dest.	1.31	4.1	1	1.7

conventional scenarios, as seen in Table 3.6.1; and from the coding perspective, is leading to the maximization of the minimum Hamming distance between the codewords. This is because we have already seen in AWGN channels that bit-wise AE learns bit-labeling in Gray coded format, thus the maximum error between two adjacent points is 1 bit. In contrast, the conventional d-QPSK is taking place in only 2-dimensions with the addition of parity check bits using Hamming codes. Thus, end-to-end learning models can learn the d-BCM designs in n times higher dimensional space, such that no CSI knowledge is required at the NN decoder in the destination node to decode the signal.

We can compare the performance of the bit-wise and symbol-wise AE framework with the following Remark 10.

Remark 10 In Table 3.6.1, we can see that the minimum Euclidean distance, packing density, and constellation figure of merit of the points designed by the symbol-wise AE and bit-wise AE becomes $\{1.30, 4.1, 1.7\}$ and $\{1.32, 3.9, 1.8\}$, respectively, both of which are very close to each other. Thus, minimizing the binary CE in bit-wise AE and categorical CE in symbol-wise AE almost forms similar d-BCM design as a spherical code (since

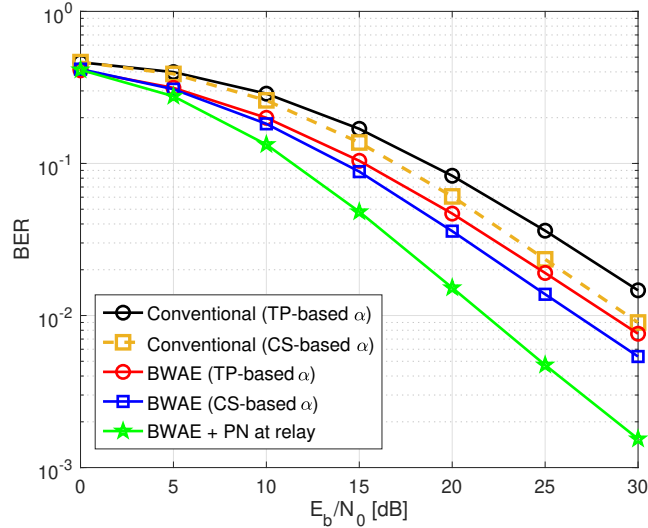


Figure 3.6.4: Bit-wise AE with power normalization (PN) at relay node.

$\chi = 1$ in both the cases). The major difference in the BER performance (≈ 5 dB as seen in Fig. 3.6.3) comes from the fact that automatic bit-labeling is done in almost a gray coded format in bit-wise AE.

Using Power Normalized AF Relay Node

We know that as the transmit SNR increases the noise power at the relay node decreases, thus the amplification factor defined as (3.2) will increase. But as we do not have CSI knowledge or noise variance of the links in the TP-based α in (3.5), we obtain the constant amplification factor of $\alpha = 0.707$, which remains sub-optimal in low SNR regimes but remains not accurate for high SNR regimes. However, CS-based α in (3.6) utilizes the second-order channel statistics and noise variance to determine the amplification factor. As a result, in Fig. 3.6.4 we can see that the knowledge of channel statistics and noise variance at the relay node helps in designing more accurate amplification factor, leading to ≈ 2 dB gain with CS-based α in BER performance compared to the TP-based α . As now, we have understood the importance of determining the accurate amplification factor, we wanted to create an amplification factor that satisfies the following constraints – (1) does not include any deep learning layers at the relay node, and (2) does not require noise variance or CSI knowledge or even second order channel statistics. Both of these conditions will be satisfied if we utilize a power normalization layer. Hence, we replace the process of constant amplification with a P_N layer as discussed in Section 3.4.2. In Fig. 3.6.4, we can see that inclusion of the P_N layer brings performance improvement of ≈ 8.5 dB over the conventional scenario with TP-based α . This is because the P_N layer is helping in normalizing the n symbols' power to n , proving extremely beneficial especially for higher SNR regimes. Interestingly, in Table 3.6.1 we can see that bit-wise

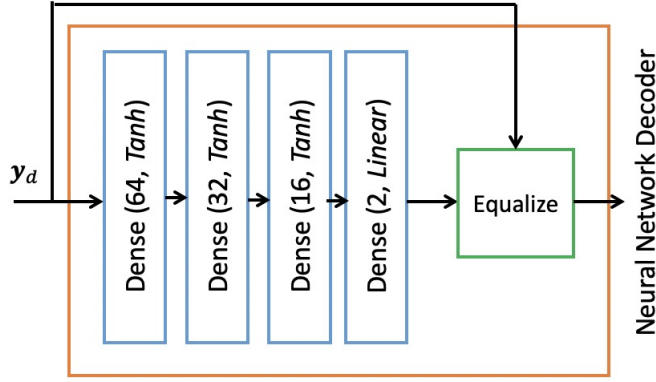


Figure 3.6.5: Radio transformer network (RTN)

AE with power normalization layer at relay forms spherical codes with $\chi = 1$ and slightly worsen the AE-based d-BCM design by reducing $d_{E_{\min}}$ and increasing E_n compared to conventional relay-based bit-wise AE. This indicates that the performance improvement by adding \mathbf{P}_N layer is only because the PN layer is creating a better amplification factor at the relay node than designed with the CS or TP-based α .

Including an Additional RTN in Lambda Layers at the Destination Node

We propose a RTN (detailed in Section 2.3) as shown in Fig. 3.6.5, and evaluate the impact of including an RTN with the NN decoder in a bit-wise AE for the AF relaying (having cascaded channels) in Fig. 3.6.6. In particular, we include RTN in Lambda layers in the NN decoder in Table 3.4.4. Including an RTN in the decoder of the proposed bit-wise AE gives the same BER performance as without an RTN in the decoder. Intuitively, this might be because we have cascaded channels from two-hops in AF relaying that needs to be decoded together, but also because even without RTN the decoder was able to decode the signals with higher accuracy so including an RTN is not helpful to improve the performance in AF relaying networks. In fact, from Table 3.6.1, we can see that including an RTN at the destination node slightly worsen the AE-based d-BCM design.

We check the convergence of the training accuracy of the bit-wise AE with and without an RTN in the decoder in Fig. 3.6.7. The RTN in decoder starts with higher accuracy (or lower CE loss), and starts converging in 2 epochs, whereas if we do not have an RTN in decoder it starts with lower accuracy and needs approximately 7 epochs for convergence. We can see that including RTN will give the same accuracy as without RTN once 15 epochs are reached. Thus, RTN in destination node for the AF relay networks can be helpful in scenarios where re-training time plays an important role in deciding the deployment of the AE network in real-world scenarios, but at the expense of slightly worse AE-based d-BCM design.

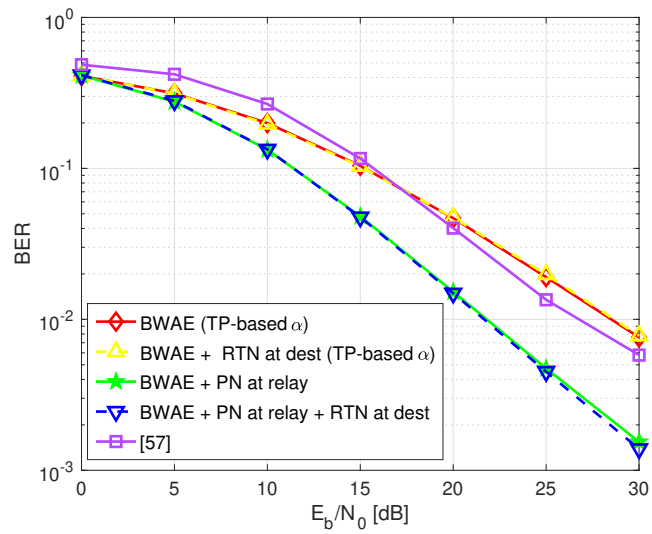


Figure 3.6.6: RTN network in decoders and comparison with [57].

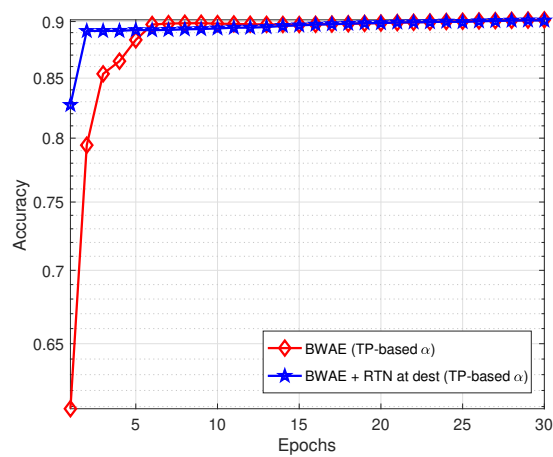


Figure 3.6.7: Convergence of BWAE in presence of RTN in decoders.

Comparison of Proposed BWAEE with [57]

All prior works on the AE-based relay networks focus on either the two-way AF relay networks performing physical network coding [38] or the decode-and-forward relay networks [37, 39, 36, 67]. Specifically, both relaying networks assume processing capabilities in the relay node, which are replaced by the NN-based architecture. However, in practice, the one-way AF relaying networks are employed widely because of their low implementation complexity. This is because all the operations, i.e., signal reception, amplification, and re-transmission, take place in the analog domain. However, in the absence of the NN-based processing at the relay node, thus removal of noise amplification cannot take place, and all the burden of the performance improvement comes to the coded-modulation designed (decoded) at the source (destination) node. In our preliminary work [57] for a one-way AF relay network, we considered the NN-based processing at the relay node. However, in this work, for the first time, we consider that the relay node does not have any NN like the conventional scenario. Thus, we compare the proposed bit-wise AE with the state-of-the-art symbol-wise AE-based AF relay network in [57] in Fig. 3.6.6. The authors in [57] utilized a NN-based relay node. In the proposed bit-wise AE, we utilize a conventional relay node with TP-based α in (3.5). Still, the proposed bit-wise AE with constant amplification factor performs better than [57] for up to $E_b/N_0 \leq 17$ dB and BER performance remains close thereafter. Further, if we utilize a power-normalization layer at the relay node, the proposed bit-wise AE always outperforms [57]. This is because in Table 3.6.1 we can see that the minimum Euclidean distance, second order moment and CFM of the proposed bit-wise and symbol-wise AE frameworks is $\{(1.32, 3.9, 1.8), (1.30, 4.1, 1.7)\}$ which is much better compared to the symbol-wise AE framework designed in [57] $\{0.9, 8.6, 0.8\}$. Thus, utilizing a conventional relay node with a power normalization layer is better than utilizing NN-based relay node, but also utilizing conventional AF relay node with TP-based α gives better performance compared to NN-based relay node up to moderate SNR, without utilizing any channel statistics or noise variance at the relay node. However, if we utilize CS-based α then a conventional AF relay node always outperforms the NN-based relay node.

3.6.3 AE-based BCM Design under RBF channels

A major concern with previous works in [57, 38] was that the AF relay node had the unfair advantage of knowing the full CSI knowledge (thus additional phase information) and presence of deep learning layers (thus additional processing-power) at the relay node. In this subsection, we consider that the relay node only knows channel gains and noise variance information, and has no deep learning layers. Also, destination node has full CSI knowledge. We keep $(n, k) = (7, 8)$. For the conventional scenario, we consider $(7, 4)$

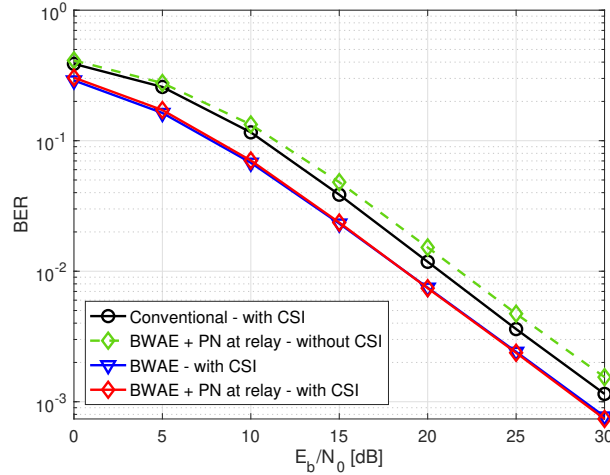


Figure 3.6.8: AF relay networks with CSI knowledge.

Hamming coding, along with QPSK modulation-demodulation and MLD decoding. We first detail the configuration of two Lambda layers for the decoder (in destination node) of the bit-wise AE with CSI knowledge – (1) in the first lambda layer, we perform channel equalization for the first-hop channel h_{sr} on received signal y_d , (2) in the second lambda layer, we perform channel equalization for the second-hop channel h_{rd} on the output of first lambda layer. Now the output after the second step is processed output of Lambda layers L_L , given to the decoder in the destination node to predict the output e_s .

In Fig. 3.6.8, we see that as E_b/N_0 increases the BER reduces. Again, similar to the reasons mentioned in Remark 9, bit-wise AE-based BCM design performs better than the conventional scenario by ≈ 3 dB. We also evaluated the performance of the proposed bit-wise AE-based BCM design with a power-normalization layer, instead of the amplification factor, at the AF relay node. Unlike, the differential scenario in Fig. 3.6.6, we see that BER performance remains the same with the PN relay, this is because we have varying amplification factor as detailed in (3.2) for conventional AF relays. Furthermore, in Table 3.6.2 we can see that spherical codes are formed ($\chi = 1$). Also, the bit-wise AE with conventional relay node obtains the superior d-BCM design compared to the relay node with a power normalization layer, similar to the differential scenario in Table 3.6.1. Lastly, for sake of comparison to the differential scenario shown in Fig. 3.6.6, we can see that even without CSI knowledge the proposed bit-wise AE with a power-normalization layer performs only 1 dB worse than conventional with CSI knowledge. Also, comparing the AE-based d-BCM design and BCM design using the Table 3.6.1 and Table 3.6.2, we can say that by using the CSI knowledge AE can design the BCM by reducing the packing density to 2.3 compared to 3.9 in the d-BCM, whereas the AE-based d-BCM is only short of 0.4 packing density compared to the conventional scenario with CSI knowledge.

Table 3.6.2: Insights for BCM design with CSI knowledge.

<i>Metrics</i>	<i>Conventional QPSK + Hamming Code</i>	<i>Bit-wise AE</i>	<i>Bit-wise AE + PN at relay</i>
Minimum Euclidean Distance $d_{E_{\min}}$ (\uparrow better)	1.4	1.8	1.7
Normalized second moment E_n (\downarrow better)	3.5	2.3	2.5
Normalized fourth moment χ	1	1	1
Constellation Figure of Merit CFM (\uparrow better)	2	3.1	2.9

Table 3.6.3: Total parameters in the NN-based encoder, decoder and RTN.

NN Node	Total Parameters	
	<i>BWAE</i>	<i>SWAE</i>
<i>NN-based Encoder</i>	3,422	19,294
<i>NN-based Decoder</i>	47,336	55,520
<i>NN-based RTN</i>	3,488	3,488

3.6.4 Computational Complexity and Time-Cost Analysis

In this subsection, we detail the computational complexity and time-cost analysis for the BCM and d-BCM designs for all the proposed AE-based end-to-end learning frameworks below

- *Total Number of Parameters* – We consider dense layers in this chapter, which have associated weights $\mathbf{W}_l \in \mathbb{R}^{\delta_{l-1} \times \delta_l}$ and bias $\mathbf{r}_l \in \mathbb{R}^{\delta_l}$ terms as optimization parameters for each l^{th} dense layers. The total number of parameters in each of the proposed NN-based encoder, decoder and RTN (for both the BWAE and SWAE) can be given as

$$P = \sum_{l=0}^J \delta_l \times \delta_{l+1} + \sum_{l=1}^J \delta_l \quad (3.20)$$

where for the NN-based encoder we have $J = 4$, $\delta_0 = k$ in BWAE, $\delta_0 = 2^k$ in SWAE; for the NN-based decoder we have $J = 5$, $\delta_5 = k$ in BWAE, $\delta_5 = 2^k$ in SWAE; and for the NN-based RTN we have $J = 4$, $\delta_0 = 2n$, respectively. Furthermore, the power normalization layer and Lambda layer in the proposed AE-based frameworks does not have any optimization parameters. For sake of summary, we provide the total parameters in the NN-based encoder, decoder and RTN in Table 3.6.3.

Directly, the total number of optimization parameters in BWAE and SWAE can be

Table 3.6.4: Time-cost analysis.

Scenario	AE-based Model	Training Cost (in sec.)	Testing Cost (in $\times 10^{-5}$ sec.)
<i>d</i> -BCM	BWAE (TP-based α)	33.2	3.7
<i>d</i> -BCM	BWAE (CS-based α)	33.2	3.7
<i>d</i> -BCM	BWAE + PN at relay	33.2	3.7
<i>d</i> -BCM	BWAE + RTN at dest	39.4	4.2
<i>d</i> -BCM	BWAE + PN at relay + RTN at dest	39.7	4.2
<i>d</i> -BCM	SWAE (TP-based α)	49.6	4.7
<i>d</i> -BCM	[57]	251.4	6.1
BCM	BWAE	137.3	4.8
BCM	BWAE + PN at relay	137.2	4.8

calculated as the sum of individual optimization parameters in Table 3.6.3, which is much less, especially for a BWAE, than the AE-based AF relay network proposed in [57] where the total number of parameters are 114,286.

- *Memory Space* – The memory space of the proposed AE-based frameworks directly depends on the total optimization parameters in the NN (detailed above). However in this chapter, we propose to utilize either a conventional or a PN-based AF relay node. As detailed in Table 6.2.1, the conventional AF relay does not require a memory buffer since it amplifies and re-transmit the received signal at each time-instant. However as detailed in Table 3.4.3, the PN-based AF relay requires a memory buffer to store n symbols to perform power normalization. Thus, utilizing a PN-based AF relay has a higher memory cost compared to the conventional AF relay node.
- *Training and Testing Cost* – In this chapter, we utilize an Intel Core i7-6700 CPU, with a GeForce RTX 2080 Ti GPU of 11 GB RAM on an Ubuntu 18.04.4 LTS OS to run our simulations both during the training and testing phase. We detail the training and testing time-cost for the *d*-BCM and BCM designs in Table 3.6.4, wherein we utilize $\approx 41\%$ and $\approx 36\%$ of the GPU during training and testing, respectively. In Table 3.6.4, we can make the following observations:
 - The training and testing cost of AEs performing *d*-BCM design is lesser compared to AEs performing BCM design, this is because in the BCM design the proposed AE-based frameworks additionally include Lambda layers at the NN decoder that performs the channel equalization.
 - BWAE takes lesser training time in comparison to the SWAE (even though they both have the same NN architectures) this is because BWAE has k input-output compared to 2^k input-output in SWAE leading to a larger number of parameters (as detailed in Table 3.6.3) in SWAE.

- BWAE takes lesser testing time in comparison to the SWAE this is because we have an additional step of performing heuristic-bit labeling in the SWAE.
- Including a PN layer at the relay node does not impact on training-testing time cost of the AE-based frameworks.
- In the d-BCM design, the RTN at the destination node increases the NN parameters in the AE (as detailed in Table 3.6.3) thereby increasing the training and testing time cost.
- The proposed AE frameworks take almost $6.6\times$ lesser training time and 39% lesser testing time in comparison to the AE-based AF relay networks in [57].

3.7 Conclusion

In this chapter, we propose end-to-end learning-based BCM and d-BCM designs in $2n$ -dimensional space using the bit-wise and symbol-wise AE frameworks for the AF relaying network. Also, in this chapter we propose to employ a conventional AF relay node instead of an NN-based relay node to minimize the implementation complexity and also improve BER performance compared to NN-based AF relay node in our previous work [57]. Furthermore, the proposed bit-wise AE frameworks take almost $6.6\times$ lesser training time and 39% lesser testing time in comparison to [57]. Thus, in the Chapter 4 on AF relay networks of this thesis, we propose to utilize a conventional AF relay node.

We create a single AE model trained on multiple values of SNRs, that can be deployed for any testing SNR, without the need of the SNR value for prediction. We show that the NN-based encoder forms 2^k codewords as a spherical code for both symbol-wise or bit-wise AE frameworks. Also, we show that minimizing the binary CE loss in bit-wise AE and categorical CE loss in symbol-wise AE almost forms a similar coded-modulation design as a spherical code. The major difference in the BER performance ≈ 5 dB comes from the fact that automatic bit-labeling is done in a gray-coded format in bit-wise AE, whereas we need to perform the bit-labeling in symbol-wise AE by solving a $2^k!$ combinatorial problem. We show that minimizing the binary CE loss for the bit-wise AE instead of the categorical CE loss for the symbol-wise AE leads to faster convergence of training at low SNR regimes, while both the AEs converge to the upper bound of estimated MI at a similar moderate SNR. Furthermore, we show that the bit-wise AE takes lesser training and testing time in comparison to the symbol-wise AE because of automatic bit-labeling and reduced input-output dimensions. Thus, in the rest of this thesis, we utilize the bit-wise AE framework for different communication networks.

The proposed AE frameworks (by performing d-BCM designs) are capable of decoding the signal without the CSI knowledge and noise variances of any links. Also, the

traditional AF relay network is outperformed by the proposed AE by 3 dB. Moreover, including a power normalization layer at the relay node, that normalizes the n transmit symbols' power to n helps us to improve our performance by additional 5 dB. Further, including an RTN in the decoder of the proposed bit-wise AE gives the same BER performance as without an RTN in the decoder, but can be helpful in scenarios where re-training time plays an important role in deciding the deployment of the AE network. Thus, in the rest of the thesis we include RTN while performing AE-based d-BCM designs for different communication networks. However, the design of RTN is changed in each scenario for obtaining optimal solutions.

Furthermore, utilizing a conventional relay node with a power normalization layer is better than utilizing deep learning layers (or NN) at the relay node, but also utilizing a conventional AF relay node gives similar performance as the relay node with a NN. Lastly, we show that AE-based BCM design increases the packing density by 1.5 compared to the AE-based d-BCM designs.

Chapter 4

AE-based Amplify-and-Forward Relay Networks with Hardware Impairments

4.1 Introduction

Until now, most of the autoencoder (AE) frameworks employ multiple dense layers in their neural network (NN) architectures [12]–[40] (as also detailed in Table 2.1). These dense layers form a weight matrix with the optimization parameters learned by end-to-end training of the AE by minimizing the cross-entropy (CE) loss. However, as the number of neurons in a dense layer increases, the number of optimization parameters in the NN increases exponentially (please see *Example 1* in Sec. 4.3 for details). Thus, the AE frameworks employing large dense layers-based NNs [12]–[40] have a drawback of optimizing a humongous amount of optimization parameters, at the same time, using smaller dense layers-based NNs reduces the learning capabilities of the AE framework. However, with the advent of internet-of-thing (IoT) and wireless sensor networks (WSNs) devices, reducing the optimization complexity of the AE frameworks without sacrificing the AE’s performance becomes pivotal for the practical realization of AE frameworks in small devices of the future networks.

Furthermore, all the AE-based frameworks consider an ideal network without any hardware impairments [12]–[40]. In practice, any communication systems, such as relaying networks, are compromised by the hardware impairments, e.g., in-phase (I) and quadrature-phase (Q) imbalance (IQI), power amplifier non-linearities, and phase noise, leading to undesirable effects such as crosstalks, added image signal, frequency interference, etc, deteriorating the network performance [19, 68, 69, 13, 70, 71].

In an ideal case of I/Q matching, the maximum signal-to-interference-and-noise-ratio¹ (SINR^{\max}) becomes *infinity*, while even for small 5° phase and 1 dB amplitude offsets can

¹The maximum SINR (SINR^{\max}) is calculated by considering IQI and additional hardware impairments (that has an additional noise effect), without considering any noise at the AF relay and destination nodes.

deteriorate the maximum SINR to 25 dB and 20 dB, respectively [19]. To counter such extreme SINR degradation, the zero-forcing (ZF)-based IQI compensation and optimal maximum likelihood detector (MLD) have been designed for an amplify-and-forward (AF) relaying network impacted by IQI under a moderate and high SINR regimes [19].

Apart from the IQI that have a multiplicative effect on the signal, tackling the other additional hardware impairments (AHI) (also referred as residual hardware impairments) that have an additive effect on the signal is of notable importance [13]. In the seminal work [70] the outage probability for both AF and decode-and-forward (DF) relaying is investigated by considering AHI at the source and relay nodes. While both the multiplicative IQI and additive AHI (IQI-AHI) for AF relaying is emphasized in [71] with optimum MLDs for IQI-AHI compensation.

However, any IQI compensation algorithm such as ZF-based or optimal MLDs [19], [71] requires the information about IQI parameters, which needs to be estimated separately based on the estimation of the effective channel that comprises the IQI parameters [72]. Thereby, IQI estimation leads to increased feedback overhead.

Furthermore, the channel state information (CSI) knowledge of the source-relay-destination channels at the destination node remains a prerequisite for designing IQI compensation algorithms [19, 68, 69, 13, 70, 71, 72]. Thus, differential modulation designs remains limited to an ideal AF relay network with no hardware impairments [63].

Extending the AE-based AF relay network studied for an ideal scenario with no hardware impairments in Chapter 3. In this chapter, we propose an AE-based AF relay network with NN-based source and destination nodes impacted by the hardware impairments (IQI and/or AHI), with an ideal (without any hardware impairments) AF relay node assisting the transmission. We propose the amplification factor for the AF relay in the absence of the CSI knowledge. We also analyze the destructive effects of the IQI and/or AHI on the received SINR at the destination node. The major contributions of this chapter are detailed as follows:

- In contrast to the traditional AE frameworks in [12]–[40] wherein a ‘single large’ NN decoder is considered at the destination node, hereby referred as ‘Single AE’, we propose a novel *Average AE* with ‘three small’ NN decoders (d_1, d_2, d_3) at the destination node. Each of these three decoders obtain soft probabilistic outputs, which are averaged to get the final soft output from the decoder. The benefits of the proposed Average AE are – (i) the number of combined optimization parameters of three small NN decoders remain significantly less compared to the number of parameters in a single large NN decoder, and (ii) the averaging from three soft outputs help in improving the BER performance compared to Single AE.
- We propose an Average AE-based block coded modulation (BCM) design with CSI

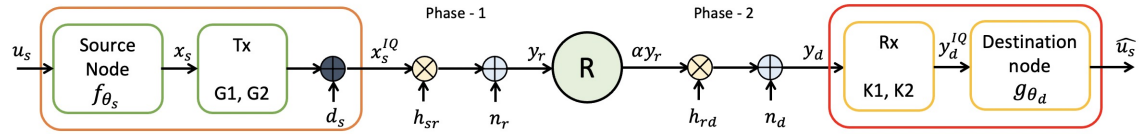


Figure 4.2.1: AF relay network with I/Q imbalance and aggregate hardware impairments.

knowledge and differential BCM (d-BCM) design without CSI knowledge for the AF relay networks in the presence of IQI and/or AHI. Within this Average AE framework, we firstly propose the BCM design with the IQI parameters and CSI knowledge, wherein we perform ZF-based IQI compensation while the NN decoders in Average AE aims to learn removal of the residual IQI and/or AHI. Secondly, we remove the necessity of the IQI parameters, by proposing BCM design with CSI knowledge and no IQI compensation, such that the Average AE aims to directly learn the removal of IQI and/or AHI. Thirdly, we remove the necessity of both the IQI parameters and CSI knowledge, by proposing d-BCM design, wherein we also design a radio transformer network (RTN) that helps in removing the impacts of channel and hardware impairments from the received signal.

- For a low SINR regime (SINR^{\max} ranging from 2.6 – 6.6 dB) we consider the AF relay network in the presence of – (i) only IQI, (ii) only AHI, (iii) both IQI-AHI, our performance results show that the Average AE-based BCM designs and d-BCM designs outperform the optimal MLD with IQI and/or AHI shifting the error floors by extraordinary margins, even outperforming ideal relay networks with no hardware impairments (IQI and/or AHI).

4.2 System Model

In this section, we detail the system model for a dual-hop AF relay network, where a source node (S) communicates with a destination node (D) using an AF relay node (R) in the presence of IQI-AHI at the source and destination nodes, as depicted in Fig. 4.2.1. All three nodes are assumed to have a single antenna. The direct link between the source and destination nodes is assumed to be highly attenuated because of large-scale path-loss or shadowing effects.

4.2.1 Modelling I/Q Imbalance

We detail the IQI at the radio frequency front-ends of the source (transmitter (Tx) side) and destination (receiver (Rx) side) nodes, while we assume the ideal AF relay transmission with no IQI [19]. Acknowledging the IQI effects, the complex local oscillator (LO)

signals at Tx and Rx side can be given as [19]:

$$\begin{aligned} z_T(t) &= \cos(\omega_L t) + j\xi_T \sin(\omega_L t + \phi_T) \\ &= G_1 e^{j\omega_L t} + G_2 e^{-j\omega_L t} \end{aligned} \quad (4.1)$$

$$\begin{aligned} z_R(t) &= \cos(\omega_L t) + j\xi_R \sin(\omega_L t + \phi_R) \\ &= K_1 e^{-j\omega_L t} + K_2 e^{j\omega_L t} \end{aligned} \quad (4.2)$$

where $\omega_L = 2\pi f_L$, while f_L denotes the frequency of the LO; $\{\xi_T, \phi_T\}$ and $\{\xi_R, \phi_R\}$ represent the total effective amplitude and phase imbalances of the source and destination nodes, respectively. By using (4.1), (4.2) we can obtain the IQI parameters of the source node (G_1, G_2) and destination node (K_1, K_2) as follows:

$$G_1 = \frac{1}{2} (1 + \xi_T e^{j\phi_T}), \quad G_2 = \frac{1}{2} (1 - \xi_T e^{-j\phi_T}), \quad (4.3)$$

$$K_1 = \frac{1}{2} (1 + \xi_R e^{-j\phi_R}), \quad K_2 = \frac{1}{2} (1 - \xi_R e^{j\phi_R}) \quad (4.4)$$

Please note that in the ideal case of I/Q matching, IQI parameters reduce to $\xi_T = \xi_R = 1$ and $\phi_T = \phi_R = 0^\circ$. Thus, in the ideal case, we have $G_1 = K_1 = 1$ and $G_2 = K_2 = 0$.

4.2.2 Signal Transmission Model

In the first phase, the source node takes k bits as input, represented as $\mathbf{u}_s \in \{0, 1\}^k$, and maps it to a complex baseband symbol $x_s \in \mathbb{C}$, such that $\mathbb{E}\{|x_s|^2\} = 1$. The up-converted signal in the presence of Tx side IQI-AHI at the source node [71] becomes

$$x_s^{IQ} = G_1 x_s + G_2^* x_s^* + d_s \quad (4.5)$$

where $(\cdot)^*$ denotes the conjugate operation, and d_s is a complex Gaussian random variable with zero mean, variance $\mathbb{E}\{|d_s|^2\} = \sigma_{hw}^2$ and pseudo-variance $\mathbb{E}\{d_s^2\} = 0$. The received baseband signal by the relay node can be given as

$$y_r = \sqrt{P_s} h_{sr} x_s^{IQ} + n_r \quad (4.6)$$

where P_s denotes the source transmission power, $h_{sr} \sim \mathcal{CN}(0, 1)$ is the first-hop channel and n_r is the additive white Gaussian noise (AWGN) at the relay node with zero mean and σ_r^2 variance. In second phase, the relay node amplifies the received signal with amplification factor, given as

$$\alpha = (P_s |h_{sr}|^2 \mathbb{E}\{|x_s^{IQ}|^2\} + \sigma_r^2)^{-1/2} \quad (4.7)$$

The amplified signal $x_r = \alpha y_r$ is re-transmitted and the signal received by the destination node is represented as

$$y_d = \sqrt{P_r} \alpha h_{rd} x_r + n_d \quad (4.8)$$

where P_r is the relay transmission power, $h_{rd} \sim \mathcal{CN}(0, 1)$ is the second-hop channel and $n_d \sim \mathcal{CN}(0, \sigma_d^2)$ is the AWGN at the destination node. Considering the Rx side IQI effect, we have

$$\begin{aligned} y_d^{IQ} &= K_1 y_d + K_2 y_d^* \quad (4.9) \\ &= \underbrace{\sqrt{P_s P_r} \alpha (K_1 G_1 h_{sr} h_{rd} + K_2 G_2 h_{sr}^* h_{rd}^*)}_{\text{Desired signal}} x_s + \\ &\quad \underbrace{\sqrt{P_s P_r} \alpha (K_1 G_2^* h_{sr} h_{rd} + K_2 G_1^* h_{sr}^* h_{rd}^*)}_{\text{Self-interference signal with IQI}} x_s^* + \\ &\quad \underbrace{\sqrt{P_s P_r} \alpha (K_1 h_{sr} h_{rd} d_s + K_2 h_{sr}^* h_{rd}^* d_s^*)}_{\text{AHI noise with IQI}} + \\ &\quad \underbrace{\sqrt{P_r} \alpha (K_1 h_{rd} n_r + K_2 h_{rd}^* n_r^*) + K_1 n_d + K_2 n_d^*}_{\text{Noise with IQI}} \\ &= (\Lambda x_s + \Omega x_s^*) + \tilde{n}_{d_s} + \tilde{n}_{rd} \quad (4.10) \end{aligned}$$

where

$$\begin{aligned} \Lambda &= \sqrt{P_s P_r} \alpha (K_1 G_1 h_{sr} h_{rd} + K_2 G_2 h_{sr}^* h_{rd}^*) \\ \Omega &= \sqrt{P_s P_r} \alpha (K_1 G_2^* h_{sr} h_{rd} + K_2 G_1^* h_{sr}^* h_{rd}^*) \\ \tilde{n}_{d_s} &= \sqrt{P_s P_r} \alpha (K_1 h_{sr} h_{rd} d_s + K_2 h_{sr}^* h_{rd}^* d_s^*) \\ \tilde{n}_{rd} &= \sqrt{P_r} \alpha (K_1 h_{rd} n_r + K_2 h_{rd}^* n_r^*) + K_1 n_d + K_2 n_d^* \end{aligned}$$

In the received signal (4.10), we can see that IQI leads to signal distortion, Λx_s , and causes self-interference, Ωx_s^* , while the AHI introduces additional noise \tilde{n}_{d_s} , each of which even individually deteriorates the performance of the AF relay network significantly.

Remark 11 Please note that the benefit of the considered system model is that it encompasses all the possible scenarios for modelling the hardware impairments in an AF relay network, detailed as follows:

1. AF relay networks with IQI-AHI (c.f. [71]) – As proposed above.
2. AF relay networks with only IQI (c.f. [19], [68], [69]) – Herein, we consider only the impact of IQI, by keeping $d_s = 0$ in the proposed system model.

3. *AF relay networks with only AHI (c.f. [13, 70]) – Herein, we consider only the impact of AHI, by keeping $G_1 = K_1 = 1$ and $G_2 = K_2 = 0$ in the proposed system model.*
4. *Ideal AF relay networks – Herein, we consider no hardware impairments (no IQI and/or AHI), by keeping $G_1 = K_1 = 1$, $G_2 = K_2 = 0$ and $d_s = 0$ in the proposed system model.*

4.2.3 Without the CSI knowledge

In the case of the absence of the CSI knowledge, we propose to employ traditional differential modulation-demodulation techniques at the source and destination nodes with the signal transmission-reception process as Sec. 4.2.2. While, we modify the amplification factor in (4.7) as

$$\alpha = (P_s \sigma_{sr}^2 \mathbb{E}\{|x_s^{IQ}|^2\} + \sigma_r^2)^{-1/2} \quad (4.11)$$

where $\sigma_{sr}^2 = \mathbb{E}\{|h_{sr}|^2\}$ is the long term expectation of the first hop channel. Note a similar amplification has been adopted for an *Ideal AF relay networks* with no hardware impairments (IQI and/or AHI) in [63].

4.2.4 Signal-to-interference-and-noise-ratio (SINR) at the Destination node

Now, we evaluate the destructive effect of the hardware impairments (IQI and/or AHI) on the AF relay networks. Let us assume that the transmitted symbol power is unity and $P_s = P_r = 1$. Then, using (4.10) we can determine the SINR (in dB) at destination node as follows:

$$\text{SINR} = \frac{\mathbb{E}\{|\Lambda x_s|^2\}}{\mathbb{E}\{|\Omega x_s^*|^2 + |\tilde{n}_{ds}|^2 + |\tilde{n}_{rd}|^2\}} \quad (4.12)$$

$$= \frac{|K_1|^2 |G_1|^2 + |K_2|^2 |G_2|^2}{|K_1|^2 |G_2|^2 + |K_2|^2 |G_1|^2 + (\sigma_{hw}^2 + \sigma_r^2 + \sigma_d^2) (|K_1|^2 + |K_2|^2)} \quad (4.13)$$

Now, we can determine the maximum SINR (SINR^{\max}) (in dB) at the destination node for proposed AF relay network with IQI and/or AHI by considering there are ‘no’ noise terms at the relay and destination nodes ($\sigma_r^2 = 0, \sigma_d^2 = 0$), detailed as follows:

$$\text{SINR}^{\max} = \text{SINR} (\sigma_r^2 = 0, \sigma_d^2 = 0) \quad (4.14)$$

$$= \frac{|K_1|^2 |G_1|^2 + |K_2|^2 |G_2|^2}{|K_1|^2 |G_2|^2 + |K_2|^2 |G_1|^2 + \sigma_{hw}^2 (|K_1|^2 + |K_2|^2)} \quad (4.15)$$

In Fig. 4.2.2, by assuming same IQI at the source and destination nodes, i.e., $\phi_T = \phi_R = \phi$ and $\xi_T = \xi_R = \xi$, we analyze the impact of IQI and/or AHI on the maximum SINR (SINR^{\max}) (in dB) at the destination node of an AF relay network, obtained using (4.15). In Fig. 4.2.2a, 4.2.2c, 4.2.2e, we can see that the maximum SINR for the *Ideal AF relay networks* is $\text{SINR}^{\max} = \infty$.

We also analyze the *AF relay networks with only IQI* in Fig. 4.2.2a, 4.2.2c, wherein we see that even a small amplitude offset $\xi = 0.8$ can make $\text{SINR}^{\max} = 40$ dB and a minor phase offset $\phi = 15^\circ$ can make $\text{SINR}^{\max} = 30$ dB. Moreover, we analyze the *AF relay networks with only AHI* in Fig. 4.2.2e, where we can see that even a negligible AHI $\sigma_{hw}^2 = 0.05$ can make $\text{SINR}^{\max} = 20$ dB.

In Fig. 4.2.2b, 4.2.2d, 4.2.2f, we consider the impact of both IQI-AHI on the AF relay networks. As seen in Fig. 4.2.2b, 4.2.2d, even with no IQI ($\xi = 1, \phi = 0^\circ$) the maximum SINR is limited to only 10 dB. While comparing Fig. 4.2.2e and Fig. 4.2.2f, we see that combined IQI-AHI deteriorates the maximum SINR dramatically, limiting it to only 6 dB. Therefore, it becomes pivotal to develop novel techniques for AF relay networks that can help in the successful transmission of signals in the presence of hardware impairments (IQI and/or AHI) under such *low SINR regimes*.

4.3 Proposed Average AE-based Framework for AF Relay Networks with IQI-AHI

In this section, we propose a novel *Average AE*-based framework for the AF relay network with the IQI and/or AHI, as shown in Fig. 4.3.1. We consider block-by-block encoding and decoding at the source and destination nodes, respectively, while the signal transmission in each phase takes place symbol-by-symbol as detailed in Section 4.2.2, herein n denotes the block length.

Until now, most of the AE works employ multiple dense layers in their NN architectures [12]–[40]. Please note that the simplest NN is composed of two dense layers, say of size X and Y neurons, which will form a weight matrix of $X \times Y$ optimization parameters (stored in a weight matrix \mathbf{W}).

For example (*Example 1*), if we have two NNs (NN-1, NN-2) each comprising of two dense layers, with $\{X_1, Y_1\} = \{512, 256\}$ neurons and $\{X_2, Y_2\} = \{64, 16\}$ neurons, then the number of optimization parameters in NN-1 and NN-2 becomes 131,072 and 1024, respectively. Thus, the smaller size of dense layers in NN-2 helps in reducing the number of optimization parameters exponentially, but also naturally have less processing capabilities compared to large dense layers-based NN-1.

Thus, in this chapter, we counter the reduced processing capabilities of a single small

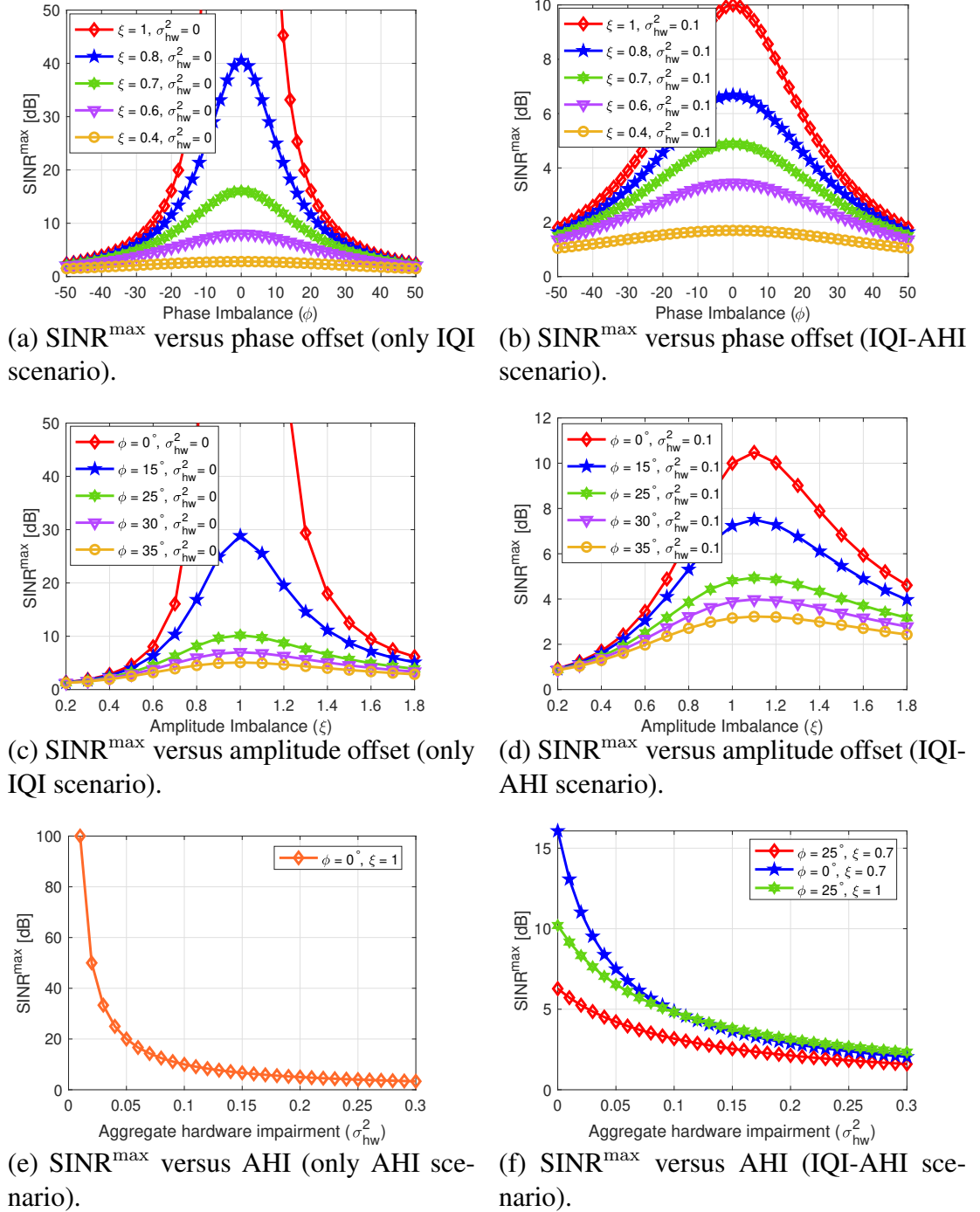


Figure 4.2.2: Impact of IQI and/or AHI on the maximum SINR SINR^{\max} (in dB).

NN decoder, by considering a collection of multiple small NN decoders. In particular, we propose that in the NN-based bit-wise AE framework, multiple small NN decoders can be learned at the same time, wherein each decoder will naturally learn different sets of weights, thereby producing different probabilistic outputs for the same input, which can be averaged to get the final output.

In particular, in contrast to the traditional AE frameworks in [12]–[40] wherein a ‘single large’ NN decoder is considered at the destination node, hereby referred as *Single AE* frameworks, we propose the *Average bit-wise AE* framework comprising of ‘three small’ NN decoders (d_1, d_2, d_3) at the destination node. We input the received signal to each of these three decoders separately. Since each of the three decoders (d_1, d_2, d_3) learn separate weight matrices, represented by \mathbf{W}^i , $i = \{1, 2, 3\}$, we obtain three individual soft probabilistic outputs from (d_1, d_2, d_3) , which are averaged to get the final soft output from the decoder. Then, we perform end-to-end training between the input bits at the encoder and the final soft output at the decoder to optimize the weight matrices at the NN encoder and decoders at the source and destination nodes, respectively.

The distinct advantages of the proposed Average AE over the Single AE in [12]–[40] are two-folds. Firstly, the small number of neurons in dense layers leads to smaller weight matrices, even with the collection of all the weights $\sum_{i=1}^3 \mathbf{W}^i$ still the number of optimization parameters is dramatically lessened compared to a single large NN-based Single AE (please see *Example 1* above). Secondly, the averaging from three different decoders’ soft probabilistic outputs enhances the decoding prowess improving the BER performance (please see *Example 2* below).

For example (*Example 2*), as shown in Fig. 4.3.1, let us consider encoding-decoding of 8 bits, where the encoder inputs 0 and 0 at its 5th and 6th bit locations. However, the decoder d_2 predicts it as 0.6 and 0.6, which in case of only a single decoder that learn weights \mathbf{W}^2 (similar to the decoder d_2) would have decoded it as 1 and 1, as in the single decoder-based Single AE in prior works [12]–[40]. But since we introduce the concept of three small decoders and taking the mean of their soft probabilistic outputs, the average soft probabilistic output becomes 0.3 and 0.4 which are decoded as 0 and 0. Naturally, averaging soft probabilistic outputs becomes even more valuable when these soft outputs lie close to the hard decision threshold of 0.5.

In this chapter, we utilize L fully-connected (dense) layers, wherein the output of the $l^{\text{th}} \in \{1, \dots, L\}$ dense layer $\omega_l \in \mathbb{R}^{\delta_l}$ can be given as (2.2). Now, we present the details of the proposed Average AE framework below.

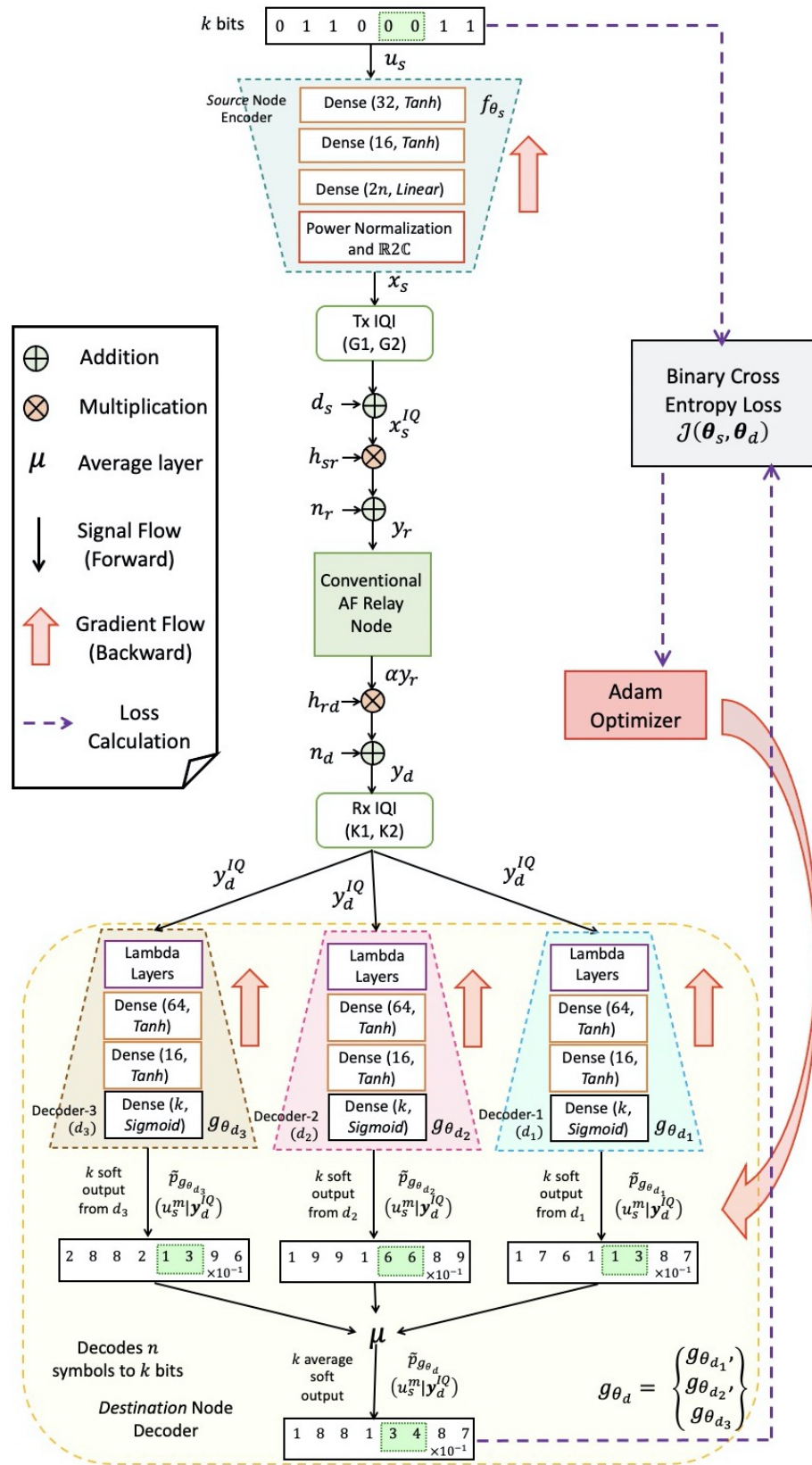


Figure 4.3.1: Proposed Average AE framework-based AF relay networks with IQI-AHI. Note the *Example 2's* 5th and 6th bits are shown in a green box.

4.3.1 Designing of the Encoder at the Source Node

The source node takes k bits $\mathbf{u}_s \in \{0, 1\}^k$ as input and maps it to n complex baseband symbols $\mathbf{x}_s \in \mathbb{C}^n$ (performing block-by-block encoding), by a mapping function $f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s)$, where θ_s is the NN encoder's parameters of the M dense layers, detailed in Definition 6 and given as (2.3). Now, the baseband representation of the up-converted signal in the presence of Tx IQI and AHI at the source node becomes $\mathbf{x}_s^{IQ} \in \mathbb{C}^n$, given as

$$\mathbf{x}_s^{IQ} = G_1 f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s) + G_2^* (f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s))^* + d_s \quad (4.16)$$

4.3.2 Designing of the AF Relay Node

The signal received at the AF relay is given by (4.6). Similar to Chapter 3, we propose to utilize the conventional AF relay node because of its low implementation complexity. Thus, the received signal is amplified as $x_r = \alpha y_r$, using the amplification factor given in, (4.7) in the presence of CSI knowledge and (4.11) in absence of the CSI knowledge.

4.3.3 Designing of the Decoder at the Destination Node

The relay re-transmits the amplified signal to the destination node, which receives the signal y_d^{IQ} as (4.10) considering the Rx side IQI effect at the destination node. Once the decoder receives n symbols, the decoder implements block-by-block decoding in each of the small decoders (d_1, d_2, d_3), such that $\mathbf{y}_d^{IQ} \in \mathbb{C}^n$ symbols are demodulated to three soft probabilistic outputs $\tilde{p}_{g_{\theta_{d_i}}}(u_s^m | \mathbf{y}_d^{IQ})$, $\forall m = 1, \dots, k$ and $i = 1, \dots, 3$, by three de-mapping functions $g_{\theta_{d_i}}(\mathbf{y}_d^{IQ}, \tilde{p}_{g_{\theta_{d_i}}}(\mathbf{u}_s | \mathbf{y}_d^{IQ}))$, $\forall i$, where θ_{d_i} denotes the weights (\mathbf{W}^i) and bias (\mathbf{b}^i) terms of N dense layers in the i^{th} NN decoder (d_i), represented as follows:

$$g_{\theta_{d_i}}(\mathbf{y}_d^{IQ}, \tilde{p}_{g_{\theta_{d_i}}}(\mathbf{u}_s | \mathbf{y}_d^{IQ})) = \sigma_N^i(\mathbf{W}_N^i \sigma_{N-1}^i(\mathbf{W}_{N-1}^i \sigma_{N-2}^i(\dots \sigma_1^i(\mathbf{W}_1^i \mathbf{L}_L(\mathbf{y}_d^{IQ}) + \mathbf{b}_1^i)\dots) + \mathbf{b}_{N-1}^i) + \mathbf{b}_N^i), \quad \forall i \quad (4.17)$$

where \mathbf{L}_L denotes the Lambda layer with no trainable parameters, as detailed in Definition 10. Now, we take the average of the three probabilistic outputs from the three decoders which becomes the final (actual) output from the NN decoder, given by

$$\tilde{p}_{g_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d^{IQ}) = \mu \left(\tilde{p}_{g_{\theta_{d_i}}}(\mathbf{u}_s | \mathbf{y}_d^{IQ}), \quad \forall i \right) \quad (4.18)$$

where $\mu(\cdot)$ denotes element-by-element mean operation. For the sake of brevity, we can also collectively show all the learning parameters and de-mapping functions in the three NN decoders as $\theta_d = \{\theta_{d_1}, \theta_{d_2}, \theta_{d_3}\}$ and $g_{\theta_d} = \{g_{\theta_{d_1}}, g_{\theta_{d_2}}, g_{\theta_{d_3}}\}$, respectively.

4.3.4 Proposed Average AE-based BCM design and d-BCM design Methodologies

We propose the Average AE-based BCM design with CSI knowledge and d-BCM design without CSI knowledge by designing the Lambda layers in the three small decoders, as detailed below:

- We propose two approaches for the *BCM design* depending on the availability of IQI parameters:

- *BCM design: ZF IQIC* – The ZF-based IQI compensation is a popular technique [19]. Herein, we aim to cancel the self interference term in (4.10) and detect the transmitted signal x_s . This is achieved by concatenating the received signal y_d^{IQ} and its conjugate y_d^{IQ*} in a matrix form, as

$$\begin{aligned} \begin{bmatrix} y_d^{IQ} \\ y_d^{IQ*} \end{bmatrix} &= \begin{bmatrix} \Lambda & \Omega \\ \Omega^* & \Lambda^* \end{bmatrix} \begin{bmatrix} x_s \\ x_s^* \end{bmatrix} + \sqrt{P_s P_r} \alpha \begin{bmatrix} K_1 h_{sr} h_{rd} & K_2 h_{sr}^* h_{rd}^* \\ K_2^* h_{sr} h_{rd} & K_1^* h_{sr}^* h_{rd}^* \end{bmatrix} \begin{bmatrix} d_s \\ d_s^* \end{bmatrix} + \\ &\quad \sqrt{P_r} \alpha \begin{bmatrix} K_1 h_{rd} & K_2 h_{rd}^* \\ K_2^* h_{rd} & K_1^* h_{rd} \end{bmatrix} \begin{bmatrix} n_r \\ n_r^* \end{bmatrix} + \begin{bmatrix} K_1 & K_2 \\ K_2^* & K_1^* \end{bmatrix} \begin{bmatrix} n_d \\ n_d^* \end{bmatrix} \\ \mathbf{y} &= \mathbf{A} \mathbf{x}_s + \mathbf{B} \mathbf{d}_s + \mathbf{C} \mathbf{n}_r + \mathbf{D} \mathbf{n}_d \end{aligned} \quad (4.19)$$

Using (7.8) we can obtain the multiplication factor $\mathbf{A} = \begin{bmatrix} \Lambda & \Omega \\ \Omega^* & \Lambda^* \end{bmatrix}$. Then, we propose to perform ZF-based IQI compensation as $\hat{y}_d^{IQ} = \mathbf{A}^{-1} \times y_d^{IQ}$ in the Lambda layer \mathbf{L}_L for all the n received symbols. Thereby, each of the three small NN decoders of the proposed Average AE framework basically aims to learn the removal of residual IQI and/or AHI.

- *BCM design: No IQIC* – The ZF-based IQI compensation assumes the knowledge of IQI parameters, but in practice, we have to estimate it separately. Thus to remove the necessity of estimating IQI parameters we let the decoders of the Average AE directly learn the removal of IQI and/or AHI. Hence, we only perform channel equalization of dual-hop channels (h_{sr}, h_{rd}) in the Lambda layer \mathbf{L}_L of three small NN decoders.

- We propose Average AE-based *d-BCM design*, wherein we remove the necessity of both CSI knowledge and IQI parameters by considering an NN-based radio transformer network (RTN) (detailed in Section 2.3) in the Lambda layers \mathbf{L}_L of each the three small NN decoders. Until now we investigated RTN to remove the channel impairments (Chapters 2, 3), in this chapter, we propose an RTN that helps in removing the dual-hop channel as well as hardware impairments (IQI and/or AHI)

Table 4.4.1: NN architectures at different nodes in the proposed Average AE framework.

NN Encoder		i^{th} NN Decoder		RTN in Lambda layer	
Neurons	Remark	Neurons	Remark	Neurons	Remark
$\delta_1 = k$	Input (\mathbf{u}_s)	$\delta_1 = 2n$	Input (\mathbf{y}_d^{IQ})	$\delta_1 = 2n$	Input (\mathbf{y}_d^{IQ})
$\delta_2 = 32$	$\sigma_1 = \text{Tanh}$	$\delta_2 = 2n$	Lambda layer (\mathbf{L}_L)	$\delta_2 = 128$	$\sigma_1 = \text{Tanh}$
$\delta_3 = 16$	$\sigma_2 = \text{Tanh}$	$\delta_3 = 64$	$\sigma_1 = \text{Tanh}$	$\delta_3 = 32$	$\sigma_2 = \text{Tanh}$
$\delta_4 = 2n$	$\sigma_3 = \text{Linear}$	$\delta_4 = 16$	$\sigma_2 = \text{Tanh}$	$\delta_4 = 2n$	$\sigma_3 = \text{Linear}$
$\delta_5 = 2n$	Power normalization (\mathbf{P}_N)	$\delta_5 = k$	$\sigma_3 = \text{Sigmoid}$	$\delta_5 = [2n, 2n]$	Concatenate $2n$ Input and $2n$ output of previous layer
$\delta_6 = 2n$	Output (\mathbf{x}_s)	$\delta_6 = k$	Soft output ($\hat{p}_{g\theta_{d_i}}(\mathbf{u}_s \mathbf{y}_d^{IQ})$)		

from the received signal at the destination node. Further, we consider each of the three small NN decoders consists of a separate NN-based RTN because each of these three RTNs will be trained with different weights, helping in a better averaging of the soft probabilistic outputs.

Remark 12 Please note the output of the Lambda layers \mathbf{L}_L for the three small NN decoders remains the same in BCM design and becomes different in d-BCM design because we only perform mathematical operations in BCM design and employ NNs in d-BCM design, respectively.

4.4 Training of the Proposed Average AE-based Framework

Please note that we design the NN architecture at encoder-decoder, training dataset creation methodology, and hyper-parameter settings to be *generalizable* for both the Average AE-based BCM design and d-BCM design.

4.4.1 Neural Network Architecture

We detail NN architecture of the proposed Average AE in Table 7.3.1. Please note for the Average AE-based d-BCM design the output of RTN is $4n$ values which becomes output of the Lambda layer (\mathbf{L}_L) in the i^{th} NN decoder, $\forall i$. All the NNs in the proposed Average AE are implemented in Keras [43] with TensorFlow [44] as backend.

4.4.2 Training and Testing Dataset Creation

We detail the training and testing dataset creation. For simplicity, we assume $\xi = \xi_T = \xi_R$, $\phi = \phi_T = \phi_R$, and $P_s = P_r = 1$. Let $\{S_{\text{train}}, S_{\text{test}}\}$ denote the training and testing samples. We create a training dataset such that trained Average AE framework can generalize

well on all the varying testing SNRs and phase offsets / amplitude offsets / AHI / both IQI-AHI during the predictions. In particular, we keep $\{S_{\text{train}}, S_{\text{test}}\} = \{3, 5\} \times 10^4$ blocks of data. Further, S_{train} training samples are generated for each of $\mathcal{S} = \{3, 8, 13, 23, 33, 43, 53, 63\}$ dB transmit SNR. For better generalization of the Average AE over testing SNRs and phase offsets or amplitude offsets or AHI or both IQI-AHI, we fix $\{\xi, \sigma_{hw}^2\} = \{0.7, 0\}$ or $\{\phi, \sigma_{hw}^2\} = \{15^\circ, 0\}$ or $\{\phi, \xi\} = \{0^\circ, 1\}$ or $\{\phi, \xi\} = \{25^\circ, 0.4\}$ and vary $\phi = \{15^\circ, 30^\circ\}$ or $\xi = \{0.4, 0.7\}$ or $\sigma_{hw}^2 = \{0.15, 0.25\}$ or $\sigma_{hw}^2 = \{0.1, 0.2\}$, respectively, and for each SNR in \mathcal{S} we generate S_{train} training samples to create the training datasets. Then, we train Average AE frameworks using each of these training datasets and test for unseen $S_{\text{test}} = 5 \times 10^5$ blocks of data samples of varying SNRs and phase offsets / amplitude offsets / AHI / IQI-AHI.

4.4.3 Training-Testing Process and Hyper-parameter Settings

The training process of the Average AE-based AF relay network is shown in Fig. 4.3.1, which aims to maximize the chances of reconstruction of the intended signal \mathbf{u}_s by learning the mapping and demapping parameters $(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d)$ at the source and destination nodes, respectively. We train the proposed Average AE by minimizing the binary cross-entropy (CE) loss, as detailed in Definition 13, between the input bits \mathbf{u}_s at the encoder and average soft output $\tilde{p}_{g_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d^{IQ})$ at the decoder. In particular, similar to the process mentioned in Sec. 2.5.1, we train the Average AE frameworks by minimizing the binary CE loss to obtain the parameters $\{f_{\theta_s}, g_{\theta_d}\}$. To train the proposed Average AE we utilize SGD with Adam [42] optimizer, where the weights are initialized using Glorot initializer [45]. We utilize step-decay to update the learning rate. In particular, we start the training with an initial learning rate $\tau_0 = 0.002$ for the first epoch and then drop the learning rate by $\eta = 0.5$ after every $D_E = 25$ epochs (we also constraint the minimum learning rate to be $\tau_{\text{min}} = 10^{-5}$). We also utilize early stopping [46] by keeping the minimum update in validation accuracy as 10^{-4} and patience to stop as 17.

Further, the testing process remains the same as Section 2.5.2 to obtain hard decision bits $\hat{\mathbf{u}}_s$ from soft probabilistic outputs.

4.5 Numerical Results

In this section, we evaluate the proposed Average AE framework for the AF relay networks with hardware impairments (IQI and/or AHI). We utilize Rayleigh block fading channels with $(n, k) = (7, 8)$ or rate $R = 8/7$ [bits/channel reuse], where the channel remains constant for $n = 7$ symbols and then changes randomly. Specifically, we evaluate the performance on the low SINR regime by considering the *Cases A–E, I* as shown

Table 4.5.1: Performance evaluation *Cases A–E, I*.

<i>Case</i>	<i>AF relay networks Scenario</i>	<i>Phase Offset (ϕ)</i>	<i>Amplitude Offset (ξ)</i>	<i>AHI (σ_{hw}^2)</i>	<i>Max. SINR (SINR^{\max})</i>	<i>Remarks</i>
<i>A</i>	<i>With only IQI</i>	30°	0.7	0	4.9 dB	Very high phase offset
<i>B</i>	<i>With only IQI</i>	15°	0.4	0	2.6 dB	Very high amplitude offset
<i>C</i>	<i>With IQI-AHI</i>	25°	0.7	0.1	3.1 dB	Both IQI-AHI
<i>D</i>	<i>With only AHI</i>	0°	1	0.15	6.6 dB	High AHI
<i>E</i>	<i>With only AHI</i>	0°	1	0.25	4.0 dB	Very high AHI
<i>I</i>	<i>Ideal</i>	0°	1	0	∞	No IQI and/or AHI

in Table 4.5.1. Please note for readability, we show the performance with transmit SNR (E_b/N_0), as the received SINR (as indicated by the maximum SINR (SINR^{\max}) calculated using (4.15)) becomes extremely low because of the presence of IQI and/or AHI.

The performance evaluation section can be divided into two parts – (1) Section 4.5.1 to 4.5.2 wherein we focus on evaluating the merits of proposed Average AE in contrast to Single AEs in [12]–[40], and (2) Section 4.5.3 to 4.5.9 wherein we focus on evaluating the proposed Average AE-based BCM and d-BCM designs with hardware impairments.

4.5.1 Comparison of the Proposed Average AE Framework with Benchmark AE Frameworks

In Fig. 4.5.1 we compare the Average AE-based CMD with following AE benchmarks designed for the AF relay networks [32], [73]:

- *Single AE [73]* – We use the single large NN decoder given in [73] for the Single AE-based AF relay networks, while the NN encoder is kept same as proposed Average AE framework.
- *CNN AE [32]* – We consider convolutional neural network (CNN)-based AE architecture proposed in [32], comprising of 1D convolutional, batch normalization, and activation layers in both the NN encoder and NN decoder of the AE framework.
- *Iterative decoding* – Instead of considering (d_1, d_2, d_3) connected parallel together as in the proposed Average AE, we consider iterative (serial) decoding with $y_d^{IQ} \rightarrow d_1 \rightarrow d_2 \rightarrow d_3$, where input of d_2 and d_3 becomes k soft probabilistic outputs of d_1 and d_2 , respectively, and output of d_3 becomes the final soft probabilistic output.

For fair comparison in AF relaying network, we consider training process and hyperparameter settings for all the AE frameworks same as Sec. 4.4.3. In Fig. 4.5.1, we can see that as SNR increases BER reduces, except the iterative decoding that cannot decode

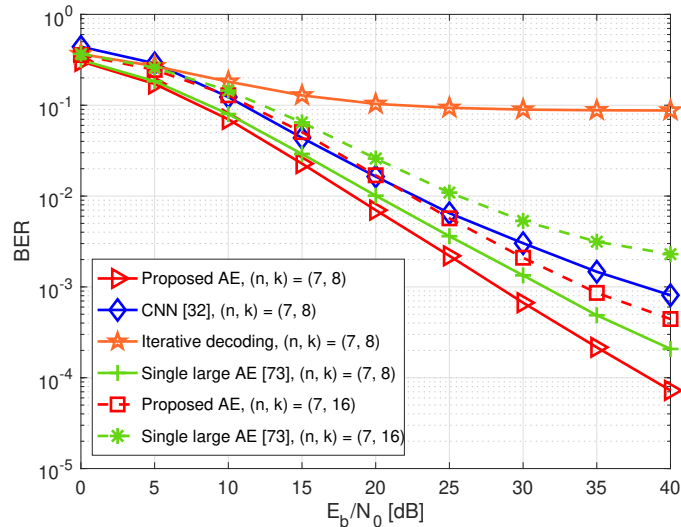


Figure 4.5.1: Comparison of the Average AE Framework with Benchmark AE Frameworks.

the signal. We observe that in the iterative decoding the first decoder performs the sub-optimal decoding because of lower NN parameters, while the later decoders are worsening the previously soft decoded outputs. Further, the Single AE [73] and Average AE based on Dense layers outperforms the CNN AE [32] based on the 1D convolutional layers. Moreover, CNN AE [32] employs a symbol-wise AE, which additionally requires solving $2^k!$ combinatorial problem for bit-labelling, compared to proposed bit-wise Average AE framework. The proposed Average AE starts performing better than the Single AE for the SNR $E_b/N_0 \geq 10$ dB. We compare the computational complexity later.

4.5.2 Higher Rate Scenario

In Fig. 4.5.1, we also consider the higher rate scenario of $(n, k) = (7, 16)$ and compare the Average AE with Single AE [73]. For higher computational prowess in Single AE [73], we also double its number of neurons in NN encoder and decoder compared to $(n, k) = (7, 8)$. We can see that as the rate $R = k/2n$ (where 2 is because of dual-hop) increases from $R = 8/14$ to $R = 16/14$ the BER performance gains achieved by the proposed Average AE over the Single AE further improves. Also, the computational complexity for $R = 16/7$ reduces by a factor of 14.6.

4.5.3 Comparison of the Proposed Average AE Framework with Benchmark AE Frameworks

In Fig. 4.5.2 for *Case C*, we compare the Average AE-based BCM design with following AE benchmarks designed for the AF relay networks [27], [57], [73] (all w/ CSI knowl-

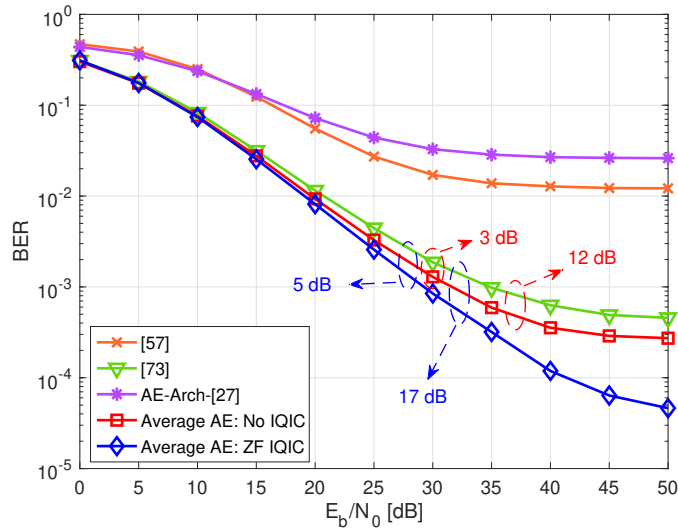


Figure 4.5.2: Comparison of the Proposed Average AE Framework with benchmark AE frameworks [27], [57], [73] (in Case C).

edge):

- [57] – A symbol-wise AE-based BCM design is designed for an ideal AF relay networks with NN-based AF relay node.
- [73] – A bit-wise AE-based BCM design is designed for an ideal AF relay network with conventional AF relay node.
- *AE architecture of [27] (AE-Arch-[27])* – Herein, we utilize the bit-wise AE architecture proposed for P2P networks in [27] for modulation design and perform (7, 4) Hamming code separately on the designed modulations, while also providing SNR information to the encoder-decoder of the AE. As [27] focussed on the P2P network; for a fair comparison, for the AF relay network, we perform training using our training dataset and hyper-parameter settings.

BER Evaluation

In Fig. 4.5.2, we compare the proposed Average AE with [27], [57], [73]. The *AE-Arch-[27]* performs channel coding and AE-based modulation design separately thus it performs the worst even with additional SNR information (compared to Average AE, [57], [73]). Although, [57] outperforms *AE-Arch-[27]* because it performs joint coded modulation design. But, [57] performs the second worst because it employs a symbol-wise AE framework which requires heuristically solving a $2^k!$ combinatorial problem to obtain the bit-labels, and [57] also considers NN-based processing at AF relay node which worsens the decoding at the destination node. Moreover, the [73] overcomes the drawbacks of [57] by considering a conventional AF relay node and a bit-wise AE framework with

automatic bit-labeling, thereby [73] outperforms [57]. However, [73] performs 3 – 12 dB worse than the proposed Average AE without IQI compensation because [73] considers a single large NN decoder, while the Average AE considers the average of soft probabilistic output from the three small NN decoders. Furthermore, we also propose Average AE with ZF-based IQI compensation, as IQI compensation has never been performed in [27], [57], [73], we can see that ZF-based IQI compensation leads to more than 17 dB gains compared to [73] at high SNR, with the knowledge of IQI parameters.

Computational Complexity Analysis

The total number of optimization parameters in the AE (encoder + decoder(s)) for – (i) [57] is 114, 286 (9, 200 + 105, 086), (ii) [73] is 50, 758 (3, 422 + 47, 336), (iii) AE-Arch-[27] is 17, 376 (88 + 17, 288), and (iv) proposed Average AE: No IQIC is 7, 462 (1, 054 + 6, 408). Thus, the proposed Average AE with three small NN decoders reduces the computational complexity by a factor of 15.3, 6.8, 2.3 compared to [57], [73] and AE-Arch-[27]² each using a single large NN decoder, respectively. Please note even for proposed Average AE-based BCM design with ZF-based IQI compensation the number of optimization parameters remains the same because we perform ZF-based IQI compensation in Lambda layer with no trainable parameters, and the proposed Average AE-based d-BCM design has additional RTN in each of the three small NN decoders, with 3, 032 optimization parameters in each RTN.

4.5.4 Benchmark Algorithms for Conventional AF Relay Networks with IQI-AHI

For the conventional scenarios, in the presence of CSI knowledge, we utilize QPSK with (7, 4) Hamming codes and consider the following as benchmarks:

- *Optimal MLD w/ CSI* – The optimal MLD in [19] performs only IQI compensation by modelling it as improper Gaussian noise. Using a similar approach, we straightforwardly obtain the optimal MLD performing IQI-AHI compensation.
- *MLD w/ CSI* – Herein, we assume we do not have the IQI parameters, thus, we use a traditional MLD without any IQI compensation.
- *Ideal MLD w/ CSI* – Herein, we consider the *Case I* (ideal AF relay network). As there are no hardware impairments we directly employ a traditional MLD.

²Although AE-Arch-[27] has the reduced complexity but it needs to predict n complex symbols in n times separately because it performs AE-based demodulation and channel decoding separately, leading to expensive prediction costs.

Please note for both *MLD w/ CSI* and *Ideal MLD w/ CSI*, we assume the information about the CSI knowledge (h_{sr}, h_{rd}), noise variances (σ_r^2, σ_d^2) and amplification factor (α). While the *Optimal MLD w/ CSI* additionally assumes the knowledge of IQI parameters ($K_{(\cdot)}, G_{(\cdot)}$) and AHI variance (σ_{hw}^2).

For the conventional scenario without the CSI knowledge, we utilize d-QPSK with (7, 4) Hamming codes and consider the following as benchmarks:

- *MLD w/o CSI* – Herein, we employ the traditional MLD without any IQI compensation.
- *Ideal MLD w/o CSI* – Herein, we consider the *Case I* (ideal AF relay network). As there are no hardware impairments we directly employ the traditional MLD.

Please note for both the *MLD w/o CSI* and *Ideal MLD w/o CSI*, we assume we do not have any information, neither IQI parameters nor CSI knowledge nor noise or AHI variances.

4.5.5 Average AE-based BCM design with No IQI Compensation

In Fig. 4.5.3, for *Cases A–C*, we analyze the proposed Average AE-based BCM design w/ CSI and without any IQI compensation (Average AE BCM design: No IQIC) and consider the benchmark MLD w/ CSI without any IQI compensation (MLD: No IQIC). We can see that the MLD can decode the signals in presence of high phase offset (*Case A*) but is unable to decode the signals with high amplitude offset (*Case B*) or for both IQI-AHI (*Case C*). This is because, with the presence of IQI and/or AHI, the maximum SINR (SINR^{\max}) reduces dramatically as seen in Table 4.5.1. However, the Average AE-based BCM design can decode the signals in the presence of high amplitude/phase/IQI-AHI, even outperforming the Ideal MLD with no hardware impairments for all the *Cases A, B, and C* up to E_b/N_0 as 45, 30, and 25 dB, respectively. Furthermore, compared to MLD, the proposed Average AE exceptionally improves the error floor, such as from $10^{-0.8}$ to $10^{-4.2}$ (in *Case B*) and $10^{-1.8}$ to $10^{-3.8}$ (in *Case C*). The performance gains of the Average AE-based BCM design can be explained by the following *Remark 13*:

Remark 13 *The Average AE forms 2^k codewords in the $2n$ -dimensional space, which have the normalized fourth-order moment as 1, indicating that the Spherical codes are formed. We also know that spherical codes are optimal for a small block length (n) [74]. From the modulation perspective, Average AE leads to the maximization of the minimum Euclidean distance from 1.4 in QPSK to 1.6 – 1.7 for the 2^k designed codewords by the Average AE. From the coding perspective, Average AE leads to the maximization of the minimum Hamming distance. Thereby, the proposed Average AE can remove the deteriorating effects of the hardware impairments (IQI and/or AHI) efficiently, even without the knowledge of IQI parameters and compensation.*

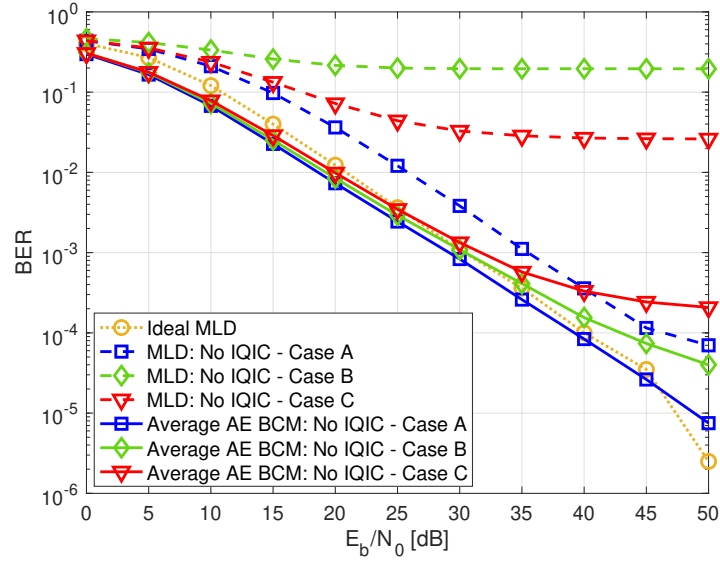


Figure 4.5.3: Average AE-based BCM design with no IQI compensation (and with CSI knowledge).

4.5.6 Average AE-based BCM design with ZF-based IQI Compensation

In Fig. 4.5.4, for *Cases A–C*, we analyze the proposed Average AE-based BCM design with ZF-based IQI compensation (Average AE BCM design: ZF IQIC) and consider the benchmark Optimal MLD w/ CSI performing optimal IQI-AHI compensation. Although, Optimal MLD has additional information about noise and AHI variance ($\sigma_r^2, \sigma_d^2, \sigma_{hw}^2$) and amplification factor (α) compared to Average AE, but still the Optimal MLD achieves error floor around $10^{-2.5}$ BER in the presence of both IQI-AHI (*Case C*). In contrast, due to similar reasons as *Remark 13* the Average AE-based BCM design is able to decode the signals in *Case C* achieving the error floor around $10^{-4.7}$ BER, even outperforming the Ideal MLD with no hardware impairments (IQI and/or AHI) for *Cases A, B* (consisting only IQI) for all SNR range and up to $E_b/N_0 = 40$ dB for *Case C* (consisting both IQI-AHI). Thus, we do not achieve an error floor for *Cases A, B*.

Furthermore, comparing the BCM design using the proposed Average AE with no IQI compensation (in Fig. 4.5.3) and with ZF-based IQI compensation (in Fig. 4.5.4), we can see that ZF-based IQI compensation helps the Average AE to improve the performance by more than 5 dB, and also helps in removing the error floor for the *Cases A, B*. Thereby, indicating that Average AE is even able to completely remove the residual IQI after the proposed ZF-based IQI compensation.

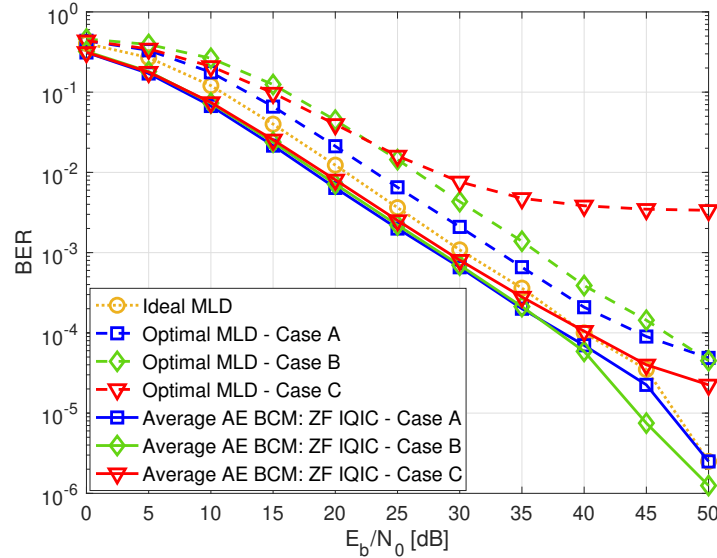


Figure 4.5.4: Average AE-based BCM design with ZF-based IQI compensation (and with CSI knowledge).

4.5.7 Average AE-based d-BCM design with No IQI Compensation

In Fig. 4.5.5, for *Cases A–C*, we analyze the proposed Average AE-based d-BCM design and consider the benchmark MLD w/o CSI, while both Average AE and MLD neither have CSI knowledge nor IQI parameters. We can see that the MLD is unable to decode the signals because we are operating under an extremely low SINR regime, and the CSI knowledge and IQI parameters is absent, thus the error floor is reached around 10^{-1} BER. However, the proposed Average AE-based d-BCM design can outperform the Ideal MLD with no hardware impairments (IQI and/or AHI) up to $E_b/N_0 = 40, 36, 30$ dB for Cases A, B, C, respectively, while reaching the error floor between 10^{-2} to 10^{-3} BER. The gains achieved by the Average AE can be understood from *Remark 13*, with the exception that in the absence of the CSI knowledge the minimum Euclidean distance increases from 0.76 for d-QPSK to 1.1 – 1.2 for 2^k codewords designed by the proposed Average AE framework in the $2n$ -dimensional space.

4.5.8 Average AE-based BCM design and d-BCM design in the presence of only AHI

In Fig. 4.5.6, we consider only the presence of AHI (*Cases D, E*), and analyze the proposed Average AE-based BCM design w/ CSI³ and d-BCM design w/o CSI, while we utilize the MLD w/ CSI and MLD w/o CSI as benchmarks, respectively. Directly, as the AHI increases the error floor increases, and the CSI knowledge helps in improving the

³Please note as no IQI is present in Cases D, E, thus we consider Average AE-based BCM design without IQI compensation.

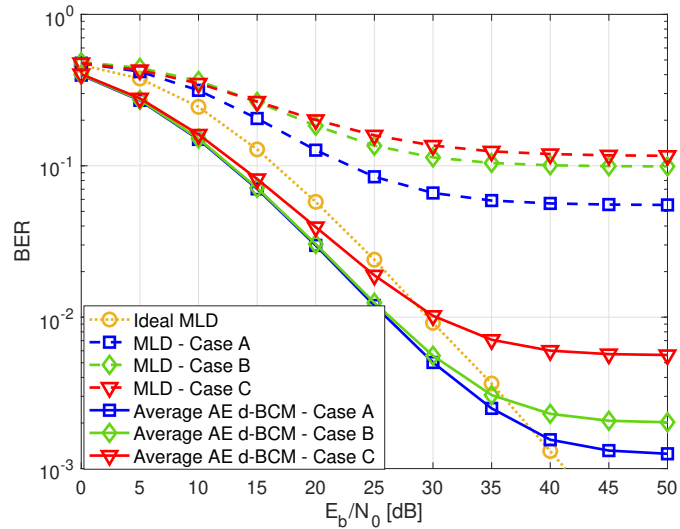


Figure 4.5.5: Average AE-based d-BCM design with no IQI compensation (and without CSI knowledge).

error floor and BER performance. Further, the presence of AHI or residual hardware impairment (*Cases C–E*) is becoming the determining factor for the error floor achieved by the Average AE. Also, the Average AE’s error floor outperforms the MLD’s error floor by a very high margin. Only when $\sigma_{hw}^2 = 0.15$ (*Case D*) the error floor for both the Average AE and MLD coincides, still the proposed Average AE outperforms the MLD in the low transmit SNRs (E_b/N_0) by 4 dB. Please note the noise floor achieved by the Average AE also indicates that it is not overfitting the training data by learning the noise in the training data, because of the considered training schedules, such as early stopping, step decay, etc (detailed in Sec. 4.4.3).

Comparing the high AHI in *Cases D, E* (Fig. 4.5.6) to moderate AHI in *Case C* (Fig. 4.5.3-4.5.5), we can also say that Average AE can remove the deteriorating impact of the AHI (or residual hardware impairments) by designing BCM design and d-BCM design intelligently (Remark 13) such that it can almost completely remove its impact up to moderate AHI $\sigma_{hw}^2 \leq 0.1$ and provide significant gains for very high AHI $\sigma_{hw}^2 > 0.1$, compared to the traditional MLD.

4.5.9 Average AE-based (d-) BCM design with Outer LDPC Codes

Until now, we have considered designing BCM design and d-BCM design for short block lengths ($n = 7$). In the recent 5G-NR standards, the outer low-density parity-check (LDPC) codes are proposed to be employed for promoting parallel implementation to satisfy the low-latency plus high-throughput conditions in 5G networks. Thus, we utilize

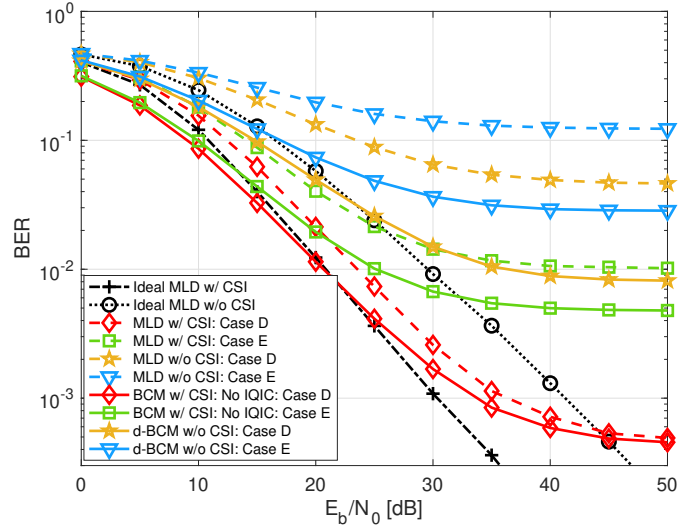


Figure 4.5.6: Average AE-based BCM design and d-BCM design in presence of only AHI.

5G-NR LDPC codes with base graph 2 (BG2)⁴ and rate⁵ $1/5$ as outer codes [75]. Specifically, we consider a block segment of $(n = 2560)$ for designing the LDPC codes with 36 filler bits. Thus, we consider *Case C* in Fig. 4.5.7, we employ 5G-NR LDPC codes as *outer codes* for the proposed Average AE-based BCM design and d-BCM design with $R = 8/7$ [bits/channel reuse], and for the conventional scenario we consider QPSK and d-QPSK in the presence and absence of the CSI knowledge with $(7, 4)$ Hamming codes and MLD detector while also employing the 5G-NR LDPC codes as *outer codes*. In Fig. 4.5.7, we can see that the gains achieved by the BCM design and d-BCM design over the MLDs for short block lengths are translated and even improved by using the outer LDPC codes, as follows:

- In Fig. 4.5.7a, we can see that the proposed Average AE-based BCM design with no IQI compensation outperforms the MLD with CSI knowledge by 8.5 dB at 10^{-4} BER.
- In Fig. 4.5.7b, we can see that the proposed Average AE-based BCM design with ZF-based IQI compensation outperforms the Optimal MLD with CSI knowledge by 7.1 dB at 10^{-4} BER.
- In Fig. 4.5.7c, we can see that the proposed Average AE-based d-BCM design without CSI knowledge or IQI parameter outperforms the MLD without CSI knowledge or IQI parameter by an extraordinary 22.1 dB at 10^{-4} BER.

⁴BG2 is employed when the information bits are in the range of 40 and 3840, while the rate lies between $1/5$ and $2/3$ [75].

⁵Please note a smaller rate gives the best chance of improving the performance for conventional MLD and reduce the performance gaps between the conventional MLD and proposed Average AE. Moreover, such smaller rates are also applied in ultra-reliable low-latency communications [76]–[77].

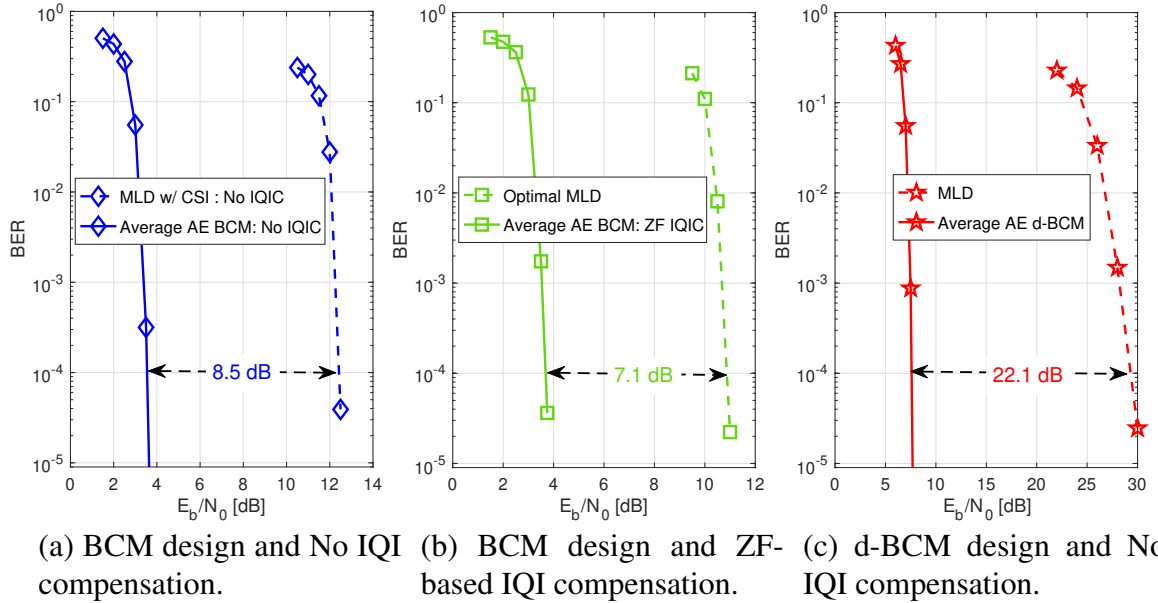


Figure 4.5.7: Comparison of Average AE-based (d-) BCM design and MLD with (7, 4) Hamming code by using LDPC as outer codes (in Case C).

This shows the significantly improved decoding capabilities by the proposed Average AE-based BCM design and d-BCM design, even when utilized for the long block lengths with the help of an outer code.

4.6 Conclusion

In this chapter, we study an AF relay network impacted by hardware impairments (multiplicative IQI and/or additive AHI). For the conventional AF relay networks we propose the amplification factor in the absence of the CSI knowledge and analyze the destructive effects of the IQI and/or AHI on the maximum received SINR at the destination nodes. We show that even individually the (IQI and/or AHI) hardware impairments dramatically reduce the maximum SINR, thereby, we focus on low SINR regimes where the maximum SINR (SINR^{\max}) at the destination node remains in the range of 2.6 dB and 6.6 dB.

Further, we propose a novel Average bit-wise AE framework with three small NN decoders instead of a single large NN decoder as in a Single AE framework. As a direct benefit the Average AE improves the BER performance by up to 3 dB while reducing the NN decoder's complexity by a factor of 7.3 in contrast to the Single AE frameworks in [12]–[40], for an ideal AF relay network with no hardware impairments. We also compare the proposed Average AE with Single AE benchmarks [27], [57], [73] in the presence of IQI-AHI. In comparison to previous best [73], the proposed Average AE outperforms by 3–12 dB without IQI compensation and more than 17 dB with the proposed ZF-based IQI compensation, while reducing the implementation complexity by a factor of 6.8.

Within the Average AE framework, we propose BCM design with CSI knowledge and ZF-based IQI compensation. Then, we show that the proposed Average AE-based BCM design effectively learns the removal of the residual IQI which helps to remove the error floors entirely, thereby also outperforming the ideal AF relay networks with no hardware impairments up to 40 dB transmit SNR. Also, in the scenarios with both IQI-AHI, the Average AE-based BCM design shifts the error floor from $10^{-2.5}$ BER in optimal MLD with IQI-AHI compensation to $10^{-4.7}$ BER, and even completely removing the error floor in the presence of only IQI impairment.

However, IQI parameter estimation is required for ZF-based IQI compensation. Thus, we remove the need for IQI parameters by proposing Average AE-based BCM design with CSI knowledge and no IQI compensation. We show that although the MLD is unable to decode the signals reaching error floor around 10^{-1} BER because of no IQI and/or AHI compensation, however, the Average AE-based BCM design can learn the IQI and/or AHI compensation without their knowledge and bring the error floor down to around 10^{-4} BER, while also outperforming the ideal MLD with no hardware impairments up to 30 dB transmit SNR.

Furthermore, to remove the necessity of both the CSI knowledge and IQI parameter information, we propose the Average AE-based d-BCM design without CSI knowledge and IQI parameter information and also design an RTN to help the Average AE to effectively estimate and remove the hardware and channel impairments. Although the traditional MLD is unable to decode the signals reaching error floors lying around 10^{-1} BER, the proposed Average AE-based d-BCM design is also able to outperform the ideal AF relay networks without IQI and/or AHI up to 30 dB transmit SNR and bring the error floor down to 10^{-3} BER.

Lastly, we consider 5G-NR LDPC codes as outer codes for the Average AE-based BCM design and can see extraordinary BER performance gains from 7.1 dB to 22.1 dB. We focus on one-way DF relaying networks with hardware impairments in Chapter 7.

Chapter 5

AE-based Two-Way

Amplify-and-Forward Relay Networks with Hardware Impairments

5.1 Introduction

We studied amplify-and-forward (AF) relay networks in Chapters 3, 4, where the source node (S) transmits a signal to the destination node (D) with the help of an AF relay node (R) in two transmission phases. Such types of relaying networks are also known as one-way (OW) AF (OW-AF) relay networks. Thus, if both the nodes want to communicate with each other, it will require four transmission phases [25, 24, 78, 79, 56, 80, 81]. However, a two-way (TW) relay network overcomes the drawbacks of the half-duplex OW relay network by utilizing the spectrum resources more efficiently. In particular, in a TW relay network, both the source and destination nodes, widely referred to as terminal nodes, communicate with each other in two transmission phases, thereby improving the spectral efficiency by 100% compared to a OW relay network [20, 21, 22, 23]. For the sake of clarity, we utilize the naming of source or destination nodes in OW relay networks as terminal nodes (A and B) and summarize the signal transmission reception between terminal nodes in OW-AF and TW-AF relay networks in Fig. 5.1.1.

In a TW-AF relay network, both the terminal nodes transmit their data simultaneously to the AF relay node, which then amplifies and re-transmit the amplified signal to the terminal nodes. Although each terminal node can perform self-interference cancellation (SIC) to remove its signal, the major challenge of a TW-AF relay network is managing the interference of simultaneously received signals at the TW-AF relay node and noise amplification with the amplification of the received signals. Thus, physical network coding techniques such as denoise-and-forward (DNF) are proposed in [65]. In the DNF technique, the relay node implements a maximum likelihood detector (MLD) on the re-

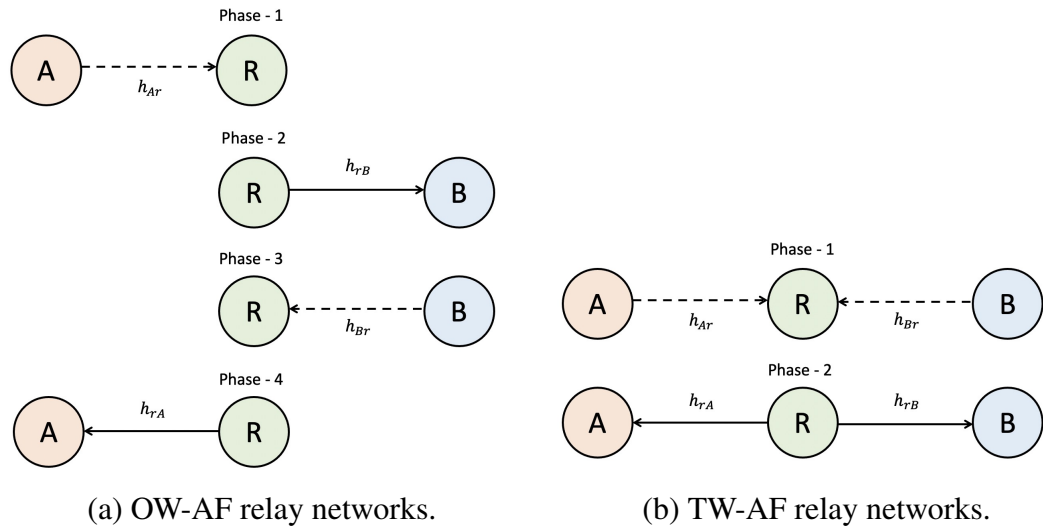


Figure 5.1.1: An illustration of OW-AF and TW-AF relay networks implementing the signal transmission–reception between terminal nodes A and B.

ceived signals from both the terminal nodes, which is followed by a denoising mapper that generates the network-coded data and a constellation mapper that modulates the signals. However, since the signals from both the terminal nodes are received by the TW-AF relay node concurrently, the MLD’s accuracy becomes very low. Moreover, the implementation complexity at the relay node also increases as the size of denoising and constellation mapper increases.

Recently, the autoencoder (AE)-based TW-AF relay networks has appeared as a potential solution to this problem [57] (our recent work), [38], [82]. The authors in [57], [38] perform AE-based block coded modulation design in fading channels, while the authors in [82] focus on only modulation design in additive white Gaussian noise (AWGN) channels. Further, [57], [38] proposed to employ dense layer-based neural network (NN) architectures, while [82] proposed a residual network-based convolutional neural network architecture. All the prior works [57, 38, 82] focus on the physical network coding where the TW-AF relay node has NN-based processing power, while none of the prior works have focussed on analog network coding, where a conventional TW-AF relay node is employed, that brings challenges of noise amplification and signal ambiguity.

As detailed in Chapter 4, in practice, relaying systems are compromised by the hardware impairments, e.g., in-phase (I) and quadrature-phase (Q) imbalance (IQI), power amplifier non-linearity, and phase noise, leading to undesirable effects such as cross-talks, an added image signal, frequency interference, etc., deteriorating the network performance [19, 68, 69, 13, 70, 71, 72]. In traditional TW-AF relay networks, the impact of IQI on a TW-AF relay network is investigated widely by considering the IQI at the terminal nodes [69], and by considering IQI at the TW-AF relay node [68]. However, none of the prior works [57, 38, 82] consider hardware impairments in their AE frameworks.

The major contributions of this chapter with specific distinctions from existing literature [38], [82] and our recent work [57] are as follows:

- We propose a Fixed bit-wise AE-based TW-AF relay network performing analog network coding impacted by IQI by jointly maximizing the bit-wise mutual information (MI) for both the terminal nodes and study the cross-entropy (CE) loss function by determining the estimated MI. We employ NN-based terminal nodes and a conventional TW-AF relay node to maintain minimal implementation cost at the relay node while having similar CSI requirements as conventional methods.
- We design an AE-based block coded-modulation (BCM) design for both the terminal nodes jointly in a $2n$ -dimensional space (n denotes block length) to tackle the interference from simultaneously received signals at the TW-AF relay node and remove the deteriorating impacts of IQI at the terminal nodes. For higher interpretability, similar to chapter 3, we study the AE-based BCM designs by using various performance metrics, such as minimum Euclidean distance, normalized second and fourth-order moments, and constellation figures of merit.
- As the NN is primarily intended for real-value processing [83], we determine the best pre-processing technique for complex-valued signals received at the NN decoder to remove the self-interference, IQI, and decode the signals.
- We consider small-scale fading with path-loss and log-normal shadow fading and determine the optimal location of the TW-AF relay node in an AE framework.

5.2 System Model

In this section, we describe the TW-AF relay network with IQI as shown in Fig. 5.2.1, where $\Gamma^{\text{th}} = \{A, B\}$ terminal node wants to communicate with the $\Upsilon^{\text{th}} = \{A, B\}$, $\Gamma \neq \Upsilon$ terminal node with the help of a TW-AF relay node (R), under the presence of both transmitter (Tx) and receiver (Rx) IQI at each of the terminal nodes. All the nodes have a single antenna, and a direct link between the terminal nodes is absent because of large-scale shadow fading and path loss effect. The rate for a TW-AF relay network is defined as $R = 2k/2n = k/n$ [bits/channel use], in contrast to $R = k/2n$ in a OW-AF relay network (in Chapters 3, 4). For the sake of clarity, we keep the block size $n = 1$ in this section. Further, the modeling of the Tx and Rx side IQI remains same as Section 4.2.1.

5.2.1 Signal Transmission Model

In the first transmission phase, each Γ^{th} terminal node transmits $\mathbf{u}_\Gamma \in \{0, 1\}^k$ bits by mapping \mathbf{u}_Γ to a complex baseband symbol $x_\Gamma = g_m(\mathbf{u}_\Gamma) \mapsto \mathbb{C}$, where $g_m(\cdot)$ denotes the

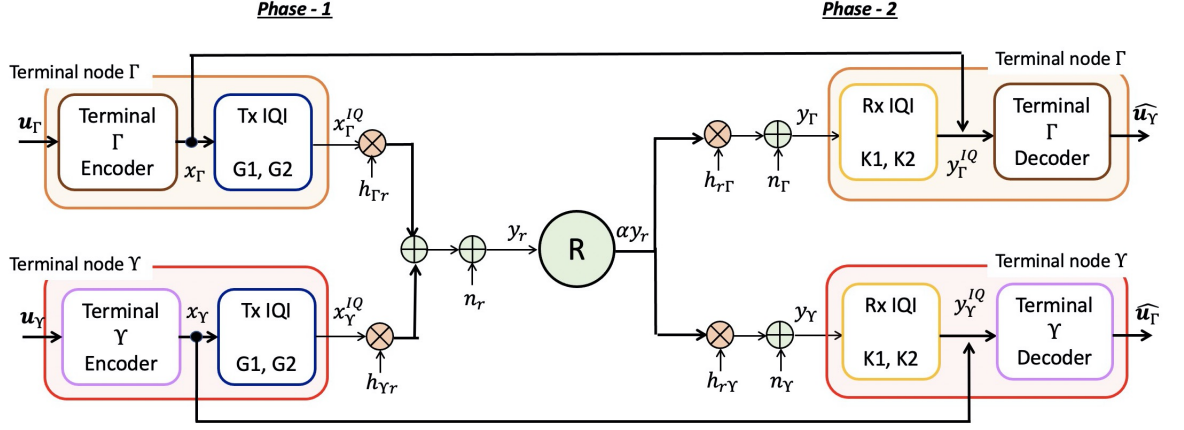


Figure 5.2.1: TW-AF Relay Network with Tx and Rx side I/Q Imbalance at the terminal nodes $\{\Gamma, \Upsilon\} = \{A, B\}$, where $\Gamma \neq \Upsilon$.

modulation process, such that $\mathbb{E}\{|x_\Gamma|^2\} = 1$. The up-converted signal in the presence of Tx side IQI is given as

$$x_\Gamma^{IQ} = G_1 x_\Gamma + G_2^* x_\Gamma^*, \quad \forall \Gamma. \quad (5.1)$$

where $(\cdot)^*$ denotes the conjugate operation. Please note that both the terminal nodes transmit their signals to the relay node together. Thus, received signal at the relay node is given by

$$y_r = \sqrt{P_A} h_{Ar} x_A^{IQ} + \sqrt{P_B} h_{Br} x_B^{IQ} + n_r \quad (5.2)$$

where P_Γ is the transmission power of the Γ^{th} terminal node, $h_{\Gamma r} \sim \mathcal{CN}(0, 1)$ is the channel between the Γ^{th} terminal node and relay node in the first transmission phase, and n_r is the additive white Gaussian noise (AWGN) at the relay node.

In the second transmission phase, the relay node amplifies the received signal with the amplification factor given as

$$\alpha = \sqrt{\frac{1}{(|G_1|^2 + |G_2|^2)(P_A |h_{Ar}|^2 + P_B |h_{Br}|^2) + \sigma_r^2}} \quad (5.3)$$

The amplified signal is broadcast to both the terminal nodes, where the signal received by the Γ^{th} terminal node is given by

$$y_\Gamma = \sqrt{P_r} h_{r\Gamma} \alpha y_r + n_\Gamma, \quad \forall \Gamma \quad (5.4)$$

where P_r is relay node's transmission power, $h_{r\Gamma} \sim \mathcal{CN}(0, 1)$ is the channel between the relay node and Γ^{th} terminal node in second transmission phase and $n_\Gamma \sim \mathcal{CN}(0, \sigma_\Gamma^2)$ is

the AWGN at the Γ^{th} terminal node. Considering the Rx side IQI effect on the received signal at the Γ^{th} terminal node, we have y_{Γ}^{IQ} as follows

$$\begin{aligned}
 y_{\Gamma}^{IQ} &= K_1 y_{\Gamma} + K_2 y_{\Gamma}^* \\
 &= \underbrace{\sqrt{P_{\Gamma} P_r} \alpha (K_1 G_1 h_{\Gamma r} h_{r\Gamma} + K_2 G_2 h_{\Gamma r}^* h_{r\Gamma}^*) x_{\Gamma}}_{\text{Own Signal with IQI}} + \\
 &\quad \underbrace{\sqrt{P_{\Gamma} P_r} \alpha (K_1 G_2^* h_{\Gamma r} h_{r\Gamma} + K_2 G_1^* h_{\Gamma r}^* h_{r\Gamma}^*) x_{\Gamma}^*}_{\text{Self-Interference of Own Signal with IQI}} + \\
 &\quad \underbrace{\sqrt{P_{\Upsilon} P_r} \alpha (K_1 G_1 h_{\Upsilon r} h_{r\Gamma} + K_2 G_2 h_{\Upsilon r}^* h_{r\Gamma}^*) x_{\Upsilon}}_{\text{Desired Signal}} + \\
 &\quad \underbrace{\sqrt{P_{\Upsilon} P_r} \alpha (K_1 G_2^* h_{\Upsilon r} h_{r\Gamma} + K_2 G_1^* h_{\Upsilon r}^* h_{r\Gamma}^*) x_{\Upsilon}^*}_{\text{Self-interference of Desired Signal with IQI}} + \\
 &\quad \underbrace{\sqrt{P_r} \alpha (K_1 h_{r\Gamma} n_r + K_2 h_{r\Gamma}^* n_r^*) + K_1 n_{\Gamma} + K_2 n_{\Gamma}^*}_{\text{Noise with IQI}} \quad (5.5)
 \end{aligned}$$

In (5.5), we can see that apart from desired signal and noise terms, each terminal node has its own signal, and self-interference from the own and desired signal terms with IQI. Thus, the terminal node Γ with the knowledge of P_{Γ} and x_{Γ} , also obtains the knowledge of $h_{\Gamma r}$, $h_{r\Gamma}$, α , P_r , and IQI parameters (K_1, G_1, K_2, G_2) . And then perform self-interference cancellation by subtracting its own signal to obtain \bar{y}_{Γ}^{IQ} , as follows

$$\begin{aligned}
 \bar{y}_{\Gamma}^{IQ} &= y_{\Gamma}^{IQ} - \sqrt{P_{\Gamma} P_r} \alpha [(K_1 G_1 h_{\Gamma r} h_{r\Gamma} + K_2 G_2 h_{\Gamma r}^* h_{r\Gamma}^*) x_{\Gamma} + \\
 &\quad (K_1 G_2^* h_{\Gamma r} h_{r\Gamma} + K_2 G_1^* h_{\Gamma r}^* h_{r\Gamma}^*) x_{\Gamma}^*] \\
 &= \sqrt{P_{\Upsilon} P_r} \alpha (K_1 G_1 h_{\Upsilon r} h_{r\Gamma} + K_2 G_2 h_{\Upsilon r}^* h_{r\Gamma}^*) x_{\Upsilon} + \sqrt{P_{\Upsilon} P_r} \alpha (K_1 G_2^* h_{\Upsilon r} h_{r\Gamma} + \\
 &\quad K_2 G_1^* h_{\Upsilon r}^* h_{r\Gamma}^*) x_{\Upsilon}^* + \sqrt{P_r} \alpha (K_1 h_{r\Gamma} n_r + K_2 h_{r\Gamma}^* n_r^*) + K_1 n_{\Gamma} + K_2 n_{\Gamma}^* \\
 &= (\Lambda_{\Upsilon\Gamma} x_{\Upsilon} + \Omega_{\Upsilon\Gamma} x_{\Upsilon}^*) + \tilde{n}_{\Gamma} \quad (5.6)
 \end{aligned}$$

where

$$\begin{aligned}
 \Lambda_{\Upsilon\Gamma} &= \sqrt{P_{\Upsilon} P_r} \alpha (K_1 G_1 h_{\Upsilon r} h_{r\Gamma} + K_2 G_2 h_{\Upsilon r}^* h_{r\Gamma}^*) \\
 \Omega_{\Upsilon\Gamma} &= \sqrt{P_{\Upsilon} P_r} \alpha (K_1 G_2^* h_{\Upsilon r} h_{r\Gamma} + K_2 G_1^* h_{\Upsilon r}^* h_{r\Gamma}^*) \\
 \tilde{n}_{\Gamma} &= \sqrt{P_r} \alpha (K_1 h_{r\Gamma} n_r + K_2 h_{r\Gamma}^* n_r^*) + K_1 n_{\Gamma} + K_2 n_{\Gamma}^*
 \end{aligned}$$

In (5.6), we can see that IQI leads to signal distortion, $\Lambda_{\Upsilon\Gamma} x_{\Upsilon}$, and causes self-interference, $\Omega_{\Upsilon\Gamma} x_{\Upsilon}^*$, which deteriorates the performance of the TW-AF relay network significantly.

Remark 14 Please note that the benefit of the considered system model is that we can also model an Ideal TW-AF relay networks with no hardware impairments (no IQI), by

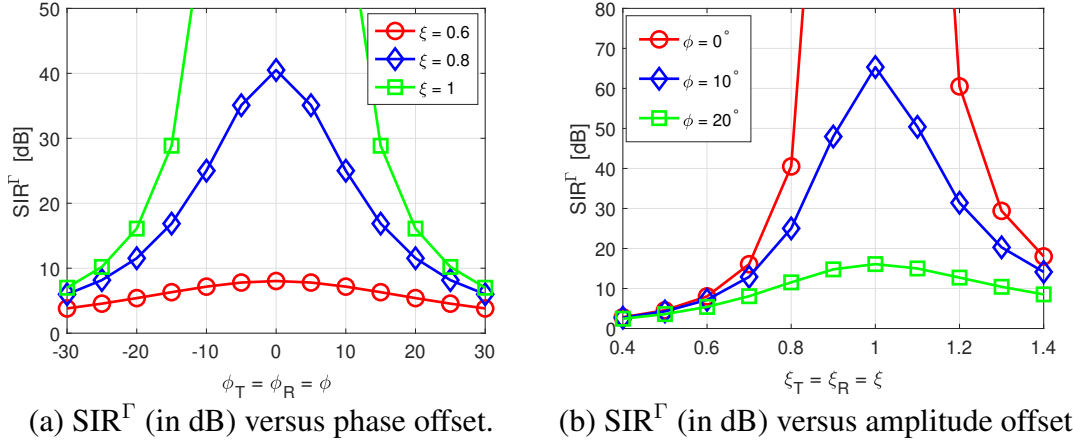


Figure 5.2.2: SIR (in dB) versus phase and amplitude offset for Γ^{th} terminal node with ($\xi_T = \xi_R = \xi$, $\phi_T = \phi_R = \phi$).

keeping $G_1 = K_1 = 1$ and $G_2 = K_2 = 0$ in the proposed system model.

5.2.2 Impact of IQI on TW-AF relay networks

Now, we assess the destructive effect of the SI term in (5.6), assuming that symbols have unit energy, and there is no noise, i.e. $n_r = n_\Gamma = 0$, then the signal-to-interference-ratio (SIR) at the Γ^{th} terminal node SIR^Γ can be given as follows

$$\begin{aligned} \text{SIR}^\Gamma &= \frac{\mathbb{E} \left\{ \left| \sqrt{P_\Upsilon P_r} \alpha (K_1 G_1 h_{\Upsilon r} h_{r\Gamma} + K_2 G_2 h_{\Upsilon r}^* h_{r\Gamma}^*) \right|^2 \right\}}{\mathbb{E} \left\{ \left| \sqrt{P_\Upsilon P_r} \alpha (K_1 G_2^* h_{\Upsilon r} h_{r\Gamma} + K_2 G_1^* h_{\Upsilon r}^* h_{r\Gamma}^*) \right|^2 \right\}} \\ &= \frac{|K_1|^2 |G_1|^2 + |K_2|^2 |G_2|^2}{|K_1|^2 |G_2|^2 + |K_2|^2 |G_1|^2} \end{aligned} \quad (5.7)$$

In Fig. 5.2.2, we analyze the impact of phase and amplitude offsets on a TW-AF relay network after the SIC from the received signal, by assuming same IQI at the terminal nodes, i.e., $\phi_T = \phi_R = \phi$ and $\xi_T = \xi_R = \xi$. Similar to chapter 4, in the case of perfect I/Q matching, SIR^Γ reach the ideal value of $\text{SIR}^\Gamma = \infty$ which is in accordance with (5.7). Also, as the phase or amplitude offset increases the SIR^Γ reduces drastically. In particular, we can see a drop of 35 dB in SIR^Γ as the amplitude offset increases from 0.8 to 0.6 for no phase-offset ($\phi = 0^\circ$) and a drop of 45 dB in SIR^Γ as the phase offset increases from 10° to 20° for no amplitude offset ($\xi = 1$). Thus, even after SIC the IQI has detrimental impacts on the SIR of the received signal at the terminal nodes in the TW-AF relay networks.

5.2.3 Designing Detector at the Terminal Node

We design the MLD with zero-forcing (ZF)-based IQI compensation (*ZF-based MLD*) for the Γ^{th} terminal node, that aims to decode the Υ^{th} node's signal x_Υ by removing the

self interference terms in (5.6). Similar to Chapter 4, this is achieved by concatenating the received signal \bar{y}_Γ^{IQ} and its conjugate \bar{y}_Γ^{IQ*} in a matrix form as

$$\begin{bmatrix} \bar{y}_\Gamma^{IQ} \\ \bar{y}_\Gamma^{IQ*} \end{bmatrix} = \begin{bmatrix} \Lambda_{\Upsilon\Gamma} & \Omega_{\Upsilon\Gamma} \\ \Omega_{\Upsilon\Gamma}^* & \Lambda_{\Upsilon\Gamma}^* \end{bmatrix} \begin{bmatrix} x_\Upsilon \\ x_\Upsilon^* \end{bmatrix} + \sqrt{P_r}\alpha \begin{bmatrix} K_1 h_{r\Gamma} & K_2 h_{r\Gamma}^* \\ K_2^* h_{r\Gamma} & K_1^* h_{r\Gamma} \end{bmatrix} \begin{bmatrix} n_r \\ n_r^* \end{bmatrix} + \begin{bmatrix} K_1 & K_2 \\ K_2^* & K_1^* \end{bmatrix} \begin{bmatrix} n_\Gamma \\ n_\Gamma^* \end{bmatrix}$$

$$\bar{\mathbf{y}}_\Gamma^{IQ} = \mathbf{A}_{\Upsilon\Gamma} \mathbf{x}_\Upsilon + \mathbf{B}_\Gamma \mathbf{n}_r + \mathbf{C}_\Gamma \mathbf{n}_\Gamma \quad (5.8)$$

Then we determine the factor $\mathbf{A}_{\Upsilon\Gamma}^{-1}$ and obtain $\mathbf{y}_\Gamma^{IQ} = \mathbf{A}_{\Upsilon\Gamma}^{-1} \times \bar{\mathbf{y}}_\Gamma^{IQ} = \mathbf{x}_\Upsilon + \mathbf{n}_\Gamma$, shown as

$$\begin{bmatrix} y_\Gamma^{IQ} \\ y_\Gamma^{IQ*} \end{bmatrix} = \begin{bmatrix} x_\Upsilon \\ x_\Upsilon^* \end{bmatrix} + \begin{bmatrix} n_\Gamma \\ n_\Gamma^* \end{bmatrix} \quad (5.9)$$

where $\mathbf{n}_\Gamma = \mathbf{A}_{\Upsilon\Gamma}^{-1} (\mathbf{B}_\Gamma \mathbf{n}_r + \mathbf{C}_\Gamma \mathbf{n}_\Gamma)$. Then, the MLD with ZF-based IQI compensation (ZF-based MLD) for the Γ^{th} terminal node is given as

$$\hat{\mathbf{u}}_\Upsilon = \arg \min_{x \in \mathcal{C}} \left\{ \left| y_\Gamma^{IQ} - x \right|^2 \right\} \quad (5.10)$$

where \mathcal{C} denotes all the possible alphabets taken by x , for example, $\pm\sqrt{1/2} \pm \sqrt{1/2}i$ (for QPSK), etc.

5.3 Proposed AE-based TW-AF Relay Network with IQI

In this section, we propose an AE-based TW-AF relay network with IQI at both the terminal nodes Tx and Rx sides as shown in Fig. 5.3.1. The first main element of the proposed AE framework is that we consider the IQI effect in the signal transmission reception. The second main element is we do not utilize any NN-based processing at the TW-AF relay node and keep the CSI requirements precisely the same as the conventional scenario. It appears as a trivial modification at first glance. However, it helps us to maintain the minimal complexity at TW-AF relays and does not have any advantage by using additional phase knowledge of the channels, unlike [57], [38].

In this chapter, we utilize L fully-connected (dense) layers, wherein the output of the $l^{\text{th}} \in \{1, \dots, L\}$ dense layer $\omega_l \in \mathbb{R}^{\delta_l}$ can be given as (2.2).

Remark 15 *Please note since we consider a TW-AF relay network in this chapter. Thus, we have a NN-based encoder and decoder pair at each terminal node. In previous Chapters 2–4, we represented the NN-based optimization parameters by $\theta_{\text{Name of the node}}$, but if we use the same notation here, it will lead to ambiguity if θ represents the NN parameter at the encoder or decoder of the node. Thus for the sake of clarity, we utilize an additional subscript Tx and Rx with the $\theta_{\text{Name of the node}}$ to denote the encoder and decoder,*

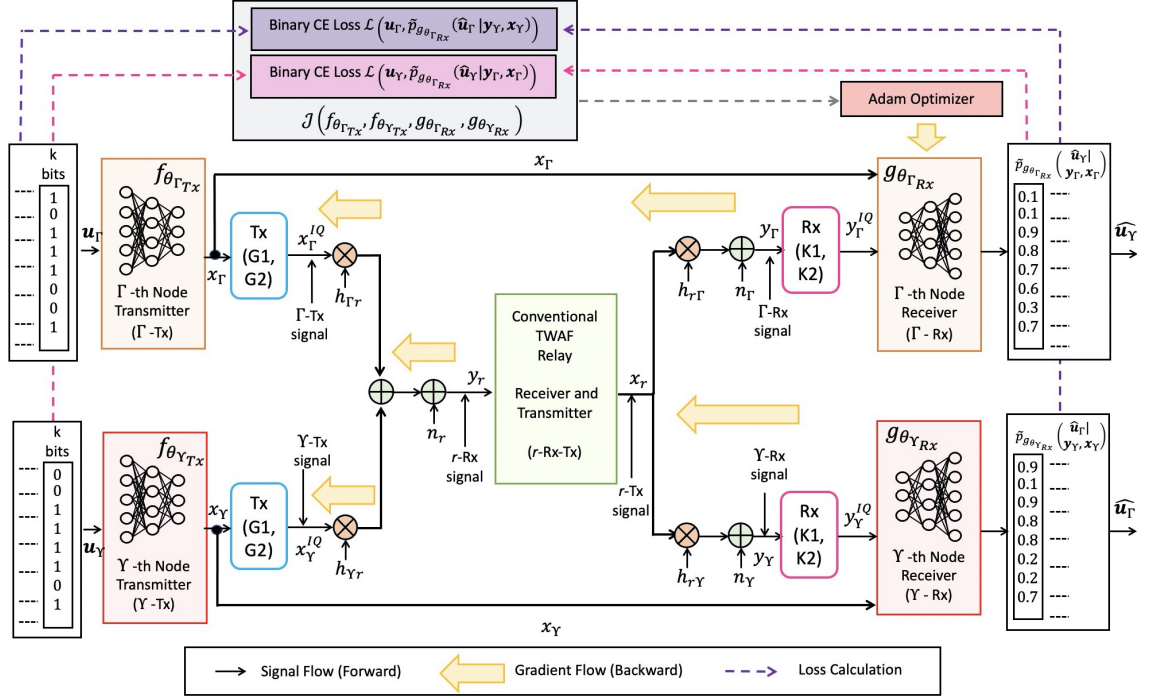


Figure 5.3.1: Proposed AE-based TW-AF relay network with IQI at both the terminal nodes Tx and Rx sides.

respectively, throughout this chapter.

5.3.1 NN-based Encoder at the Γ^{th} Terminal Node

The terminal node Γ takes k bits $\mathbf{u}_\Gamma \in \{0, 1\}^k$ as input and performs block-by-block encoding by mapping \mathbf{u}_Γ to n complex baseband symbols $\mathbf{x}_\Gamma \in \mathbb{C}^n$ using the mapping function $f_{\theta_{\Gamma Tx}}(\mathbf{u}_\Gamma, \mathbf{x}_\Gamma)$, where $\theta_{\Gamma Tx}$ is the weight and bias terms of constituent M dense layers, similar to detailed in Definition 6, and given by

$$f_{\theta_{\Gamma Tx}}(\mathbf{u}_\Gamma, \mathbf{x}_\Gamma) = \mathbf{P}_N(\sigma_M(\mathbf{W}_M \sigma_{M-1}(\dots \sigma_1(\mathbf{W}_1 \mathbf{u}_\Gamma + \mathbf{b}_1) \dots) + \mathbf{b}_M)) \quad (5.11)$$

wherein, \mathbf{P}_N denotes the power normalization layer, detailed in Definition 7, that mandates $\|\mathbf{x}_\Gamma\|_2^2 = n$.

The up-converted signal in the presence of Tx IQI is given as (5.1), and both the terminal node performs symbol-by-symbol transmission.

5.3.2 Conventional TW-AF Relay Node

The signal received by the TW-AF relay node from both the terminal nodes is given by (5.2). We propose to utilize conventional TW-AF relay node, that amplifies the received signal symbol-by-symbol using the amplification factor in (5.3) as $x_r = \alpha y_r$ and re-transmits the signal to the terminal nodes.

5.3.3 NN-based Decoder at the Γ^{th} Terminal Node

The signal received by the Γ^{th} terminal node in the presence of Rx side IQI is given by (5.5). We consider that the NN-based decoder at the Γ^{th} terminal node performs block-by-block decoding operation, thereby it takes n symbols as input ($\mathbf{y}_\Gamma^{IQ} \in \mathbb{C}^n$) and transforms it to k soft probabilistic outputs $\tilde{p}_{g_{\theta_{\Gamma_{Rx}}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \in [0, 1]^k$ of the intended signal of Υ^{th} terminal node $\Upsilon \neq \Gamma$, using a de-mapping function $g_{\theta_{\Gamma_{Rx}}}(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma; \tilde{p}_{g_{\theta_{\Gamma_{Rx}}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma))$, where $\theta_{\Gamma_{Rx}}$ denotes the weights and bias terms of constituent N dense layers at the Γ^{th} terminal node's NN decoder, given as

$$g_{\theta_{\Gamma_{Rx}}}(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma; \tilde{p}_{g_{\theta_{\Gamma_{Rx}}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)) = \sigma_N(\mathbf{W}_N \sigma_{N-1}(\mathbf{W}_{N-1} \sigma_{N-2}(\dots \sigma_1(\mathbf{W}_1 \mathbf{L}_L(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) + \mathbf{b}_1) \dots) + \mathbf{b}_{N-1}) + \mathbf{b}_N) \quad (5.12)$$

where \mathbf{L}_L denotes the Lambda layer as defined in Definition 10, with two inputs $(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)$. We will elaborate more about Lambda layer in Section 5.4. Also, please note the notation $g_{\theta_{\Gamma_{Rx}}}(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma; \tilde{p}_{g_{\theta_{\Gamma_{Rx}}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma))$ represents a decoder de-mapping function g at the Γ^{th} terminal node, that takes $\mathbf{y}_\Gamma^{IQ} \in \mathbb{C}^n$ complex baseband symbols and its own complex baseband symbols $\mathbf{x}_\Gamma \in \mathbb{C}^n$ as two inputs to give $\tilde{p}_{g_{\theta_{\Gamma_{Rx}}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \in [0, 1]^k$ soft-probabilities as output using the optimization parameter $\theta_{\Gamma_{Rx}}$.

The activation function of the N^{th} layer, σ_N , is a *Sigmoid* activation function $\sigma_N(x) = \frac{1}{1+e^{-x}}$ on each of the k logits to procure the probabilities $\tilde{p}_{g_{\theta_{\Gamma_{Rx}}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)$, $m = 1, \dots, k$ that corresponds to the log-likelihood ratios (LLRs) as:

$$\text{LLR}_\Gamma^m = \log \frac{1 - \tilde{p}_{g_{\theta_{\Gamma_{Rx}}}}(u_\Upsilon^m = 0 | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)}{\tilde{p}_{g_{\theta_{\Gamma_{Rx}}}}(u_\Upsilon^m = 0 | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)} = l_\Gamma^m, \forall \Gamma, m \quad (5.13)$$

where, $\tilde{p}_{g_{\theta_{\Gamma_{Rx}}}}(u_\Upsilon^m = 1 | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) = \sigma_N(l_\Gamma^m)$.

5.3.4 Process of Creating Training and Testing Datasets

Let us consider that there are $\{S_{\text{train}}, S_{\text{test}}\}$ training and testing samples. We elaborate on the process of creating the training-testing datasets below:

- *For ideal scenario* – We create single AE framework trained on 10^5 samples of multiple SNR values $\mathcal{S} = \{3, 10, 23, 28, 38, 42\}$ dB, that can be deployed for any testing SNR.
- *To check the impact of phase offset* – We create a training dataset such that the AE framework generalizes well for all testing SNR and phase offset. We fix $\xi_T = \xi_R = 0.8$ and create 3×10^4 samples of $\{3, 10, 23, 28, 42\}$ dB for each $\phi_T = \phi_R =$

$\{0^\circ, 10^\circ, 20^\circ\}$ to create a training dataset. We train a single AE framework using this training dataset and test it for an unseen dataset of 5×10^5 samples with varying transmit SNR and phase offsets.

- *To check the impact of amplitude offset* – We create a training dataset such that the AE framework generalizes well for all testing SNR and amplitude offset. We firstly fix $\phi_T = \phi_R = 10^\circ$ and create 3×10^4 samples of $\{3, 10, 23, 28, 42\}$ dB for each $\xi_T = \xi_R = \{0.6, 0.8, 1\}$. We train a single AE framework using this training dataset and test it for unseen datasets of 5×10^5 samples with varying transmit SNR and amplitude offsets.

5.3.5 Training and Testing Procedure

We train the proposed AE-based framework by minimizing the total binary cross-entropy (CE) loss [41] for both the terminal nodes $\Gamma \rightarrow R \rightarrow \Upsilon$ and $\Upsilon \rightarrow R \rightarrow \Gamma$ together, where $\{\Gamma, \Upsilon\} = \{A, B\}$ and $\Gamma \neq \Upsilon$ as follows:

$$\mathcal{J}(\boldsymbol{\theta}_{A_{Tx}}, \boldsymbol{\theta}_{B_{Tx}}, \boldsymbol{\theta}_{A_{Rx}}, \boldsymbol{\theta}_{B_{Rx}}) = \frac{1}{2} \sum_{\substack{\{\Gamma, \Upsilon\} = \{A, B\}, \\ \Gamma \neq \Upsilon}} \mathcal{L}_{\Upsilon R \Gamma} \left(\mathbf{u}_\Upsilon, \tilde{p}_{g_{\boldsymbol{\theta}_{\Gamma Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \right) \quad (5.14)$$

where 2 is because we are taking combined binary CE losses for transmission between $\Gamma \rightarrow R \rightarrow \Upsilon$ and $\Upsilon \rightarrow R \rightarrow \Gamma$ nodes, and the notation $\mathcal{L}_{\Upsilon R \Gamma} \left(\mathbf{u}_\Upsilon, \tilde{p}_{g_{\boldsymbol{\theta}_{\Gamma Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \right)$ represents the binary CE loss term, as defined in Definition 13, for \mathbf{u}_Υ signal transmitted by Υ terminal node and the decoded soft probability $\tilde{p}_{g_{\boldsymbol{\theta}_{\Gamma Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)$ by Γ terminal node. Now, we can estimate the total binary CE loss in (5.14) via mini-batch sampling, as follows

$$\mathcal{J}(\boldsymbol{\theta}_{A_{Tx}}, \boldsymbol{\theta}_{B_{Tx}}, \boldsymbol{\theta}_{A_{Rx}}, \boldsymbol{\theta}_{B_{Rx}}) = \frac{1}{2B^s} \sum_{q=1}^Q \sum_{b^s=1}^{B^s} \sum_{m=1}^k \sum_{\substack{\{\Gamma, \Upsilon\} = \{A, B\}, \\ \Gamma \neq \Upsilon}} \mathcal{L}_{\Upsilon R \Gamma} \left(u_\Upsilon^{(q \times b^s, m)}, \tilde{p}_{g_{\boldsymbol{\theta}_{\Gamma Rx}}} \left(u_\Upsilon^{(q \times b^s, m)} | \mathbf{y}_\Gamma^{IQ(q \times b^s)}, \mathbf{x}_\Gamma^{(q \times b^s)} \right) \right) \quad (5.15)$$

where B denotes the batch size and $Q = \frac{S_{\text{train}}}{B}$ denotes the number of training samples in a batch. The weights and bias terms in the NN are updated using the stochastic gradient descent (SGD) method using back-propagation [41]. Let us denote the NN optimization parameters as $\Theta = \{\boldsymbol{\theta}_{A_{Tx}}, \boldsymbol{\theta}_{B_{Tx}}, \boldsymbol{\theta}_{A_{Rx}}, \boldsymbol{\theta}_{B_{Rx}}\}$, then we update the weights using the Adam optimizer as detailed in Sec. 2.5.1.

For the trained AE, we perform predictions on the testing set to obtain soft probabilistic outputs $\tilde{p}_{g_{\boldsymbol{\theta}_{\Gamma Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)$ for both the terminal nodes, where $\{\Gamma, \Upsilon\} = \{A, B\}$ and $\Gamma \neq \Upsilon$. Similar to Section 2.5.2, we then perform hard decision decoding by keeping a

threshold of 0.5 to obtain the decoded bits at both the terminal nodes A and B.

5.3.6 Hyper-Parameter Setting and NN Architectures

In this chapter, we assume the same quality-of-service (QoS) requirement for both terminal nodes. Thus we keep the NN architectures the same for both the terminal nodes. Further, we employ the same NN architecture for the encoder and decoder as proposed for the bit-wise AE (BWAEE) in Chapter 3. This also shows the advantage of the proposed NN architecture, which efficiently handles both the OW-AF and TW-AF relay networks scenarios. In particular, the NN architectures for encoder and decoder is detailed in Table 3.4.1 and Table 3.4.3, respectively. To train the proposed AE network, we utilize SGD with Adam optimizer [42], where the weights of the dense layers are initialized with the Glorot initializer [45]. We keep the learning rate $\tau = 0.00125$, batch size $B = 6000$ and number of training epochs $E = 50$. Our proposed AE framework does not need the SNR value for predicting unseen testing datasets.

5.4 Simulation Results

In this section, we evaluate the performance of the AE-based framework for the TW-AF relay networks with/without I/Q imbalance. The performance evaluation is divided into two parts – (1) we consider an ideal TW-AF relay networks with no IQI in Section 5.4.1, and (2) we consider TW-AF relay networks with IQI in Section 5.4.2.

5.4.1 Ideal TW-AF Relay Networks with no IQI

In this subsection, we present the performance evaluation for an ideal TW-AF relay networks with no hardware impairments (no IQI). For the conventional scenarios, we utilize QPSK modulation and the optimal MLD at the terminal nodes.

AE-based Modulation Design under AWGN Channels

Now, we evaluate the performance under AWGN channels by considering $h_{(\cdot)} = 1$, where $(\cdot) = \{\Gamma_r, r\Gamma\}$. Please note that this assumption holds because no direct link exists between the terminal nodes. For AWGN scenario we propose to pre-process the received signal at the NN-based decoder of each Γ^{th} terminal node using three Lambda layers, given by $\mathbf{L}_L(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) = \mathbf{L}_3(\mathbf{L}_2(\mathbf{L}_1(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)))$, detailed as follows:

- L1:** Each Γ^{th} decoder removes its own signal from the received signal \mathbf{y}_Γ as $\bar{\mathbf{y}}_\Gamma = \mathbf{y}_\Gamma - \alpha \times \mathbf{x}_\Gamma$.

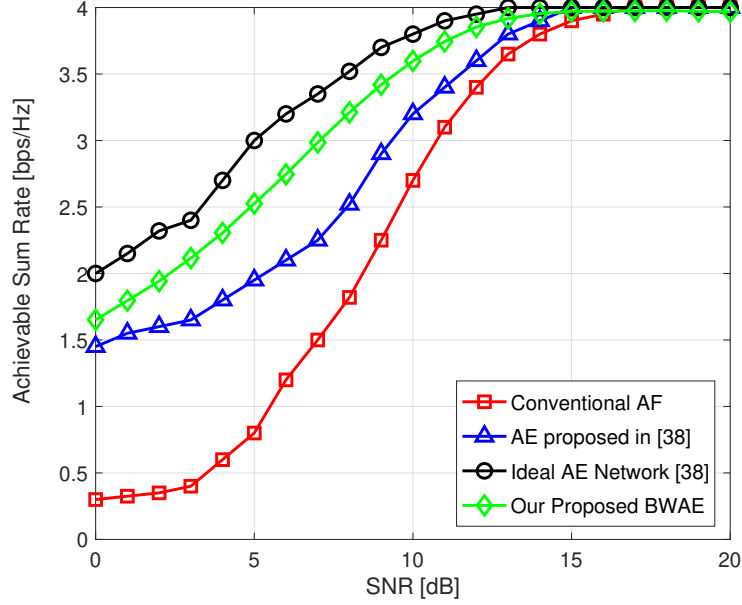


Figure 5.4.1: Comparison of Proposed BWAE v/s [38].

L2: Each Γ^{th} decoder has a *Concatenate Layer* \mathcal{C}_C , concatenating $[\mathbf{y}_\Gamma, \bar{\mathbf{y}}_\Gamma, \alpha]$. This helps us to concatenate the received signal, signal left after removing the self interference signal, and amplification factor.

L3: Each Γ^{th} decoder has a *Batch-normalization (BN)*¹ layer \mathcal{B}_N as its last lambda layer.

In [38], the authors studied AE-based TW-AF relay networks with NN-based AF relay node. They have utilized achievable sum-rate (ASR) as the performance metric given by

$$\text{ASR} = 2\varsigma (1 - \mathbb{E}(\mathcal{J}(\boldsymbol{\theta}_{A_{Tx}}, \boldsymbol{\theta}_{B_{Tx}}, \boldsymbol{\theta}_{A_{Rx}}, \boldsymbol{\theta}_{B_{Rx}}))) [\text{bps/Hz}] \quad (5.16)$$

where ς denotes number of bits per symbol and 2 is because there are two terminal nodes. We compare the performance of the proposed AE framework with [38] for $(n, k) = (1, 2)$ in Fig. 5.4.1. Where we directly utilize the Fig. 3 from [38] to plot the AE network performance of [38] and the ideal AE network performance for TW-AF relay networks. In [38] the “Ideal AE Network [38]” is assessed as an upper-bound benchmark. Herein, the relay node is considered NN-based with decoding and re-encoding power. Further, apart from the received signal, the NN relay node has information about the transmitted bits of both the terminal node $(\mathbf{u}_A, \mathbf{u}_B)$, the complex baseband symbols designed by the NN encoder of both the terminal node $(\mathbf{x}_A, \mathbf{x}_B)$, and the perfect CSI knowledge of both transmission hops.

¹In the BN layer, mean (μ_B) and variance (σ_B^2) of the samples in input batch are used to normalise the input tensor with the calculated statistics as $\boldsymbol{\omega}^{(l)} = \boldsymbol{\omega}^{(l)} - \mu_B / \sqrt{\sigma_B^2 - \epsilon}$. Finally scaled and shifted as $\boldsymbol{\omega}^{(l)} = \gamma \boldsymbol{\omega}^{(l)} + \zeta$, where $\{\gamma, \zeta\}$ are hyper-parameters learned during the training phase.

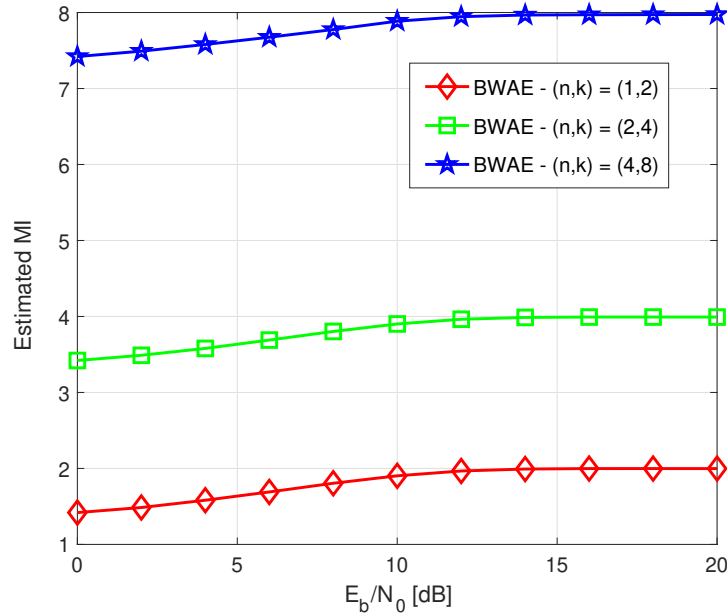


Figure 5.4.2: Estimated Mutual Information versus E_b/N_0 [dB].

For the proposed AE, we calculate the expected loss $\mathcal{J}(\boldsymbol{\theta}_{A_{Tx}}, \boldsymbol{\theta}_{B_{Tx}}, \boldsymbol{\theta}_{A_{Rx}}, \boldsymbol{\theta}_{B_{Rx}})$ using (5.14) for each transmit SNR point and calculate the ASR using (5.16). Clearly, the proposed AE performs only 2 dB worse than the ideal AE network performance (which includes an ideal NN relay node decoding the signals perfectly) and outperform the AE-network proposed in [38] for all the SNR range by 3 dB. This suggests the potential benefits of utilizing conventional TW-AF relay node over NN-based relay node, while having the benefits of lower implementation complexity and signal transmission-reception in analog-domain.

Estimated Mutual Information under RBF channels

The estimated MI is defined as Definition 14. For the TW-AF relay networks, the total estimated MI will be the summation of the MI terms for the information exchange between the terminal nodes, which takes place in two transmission hops, given by (5.24) (in Appendix 5.A). We now compare the estimated MI for the AE with $(n, k) = \{(1, 2), (2, 4), (4, 8)\}$ in Fig. 5.4.2. As the transmit SNR increases, the estimated MI increases until it reaches the upper bound k . This suggests that the KL-divergence loss approached 0 as E_b/N_0 increases, and proposed AE can well approximate at a moderate E_b/N_0 . Furthermore, for any given rate R (in Fig. 5.4.2 $R = 2$), we can see that the block size (n) does not have any impact on the convergence of the estimated MI. Moreover, in chapter 2, Section 2.7, we saw that the rate number of input bits also does not have any impact on the convergence of the estimated MI. Thus, if we have enough training samples, it ensures that the estimated MI always converges on the ‘same’ moderate SNR regime, showing the

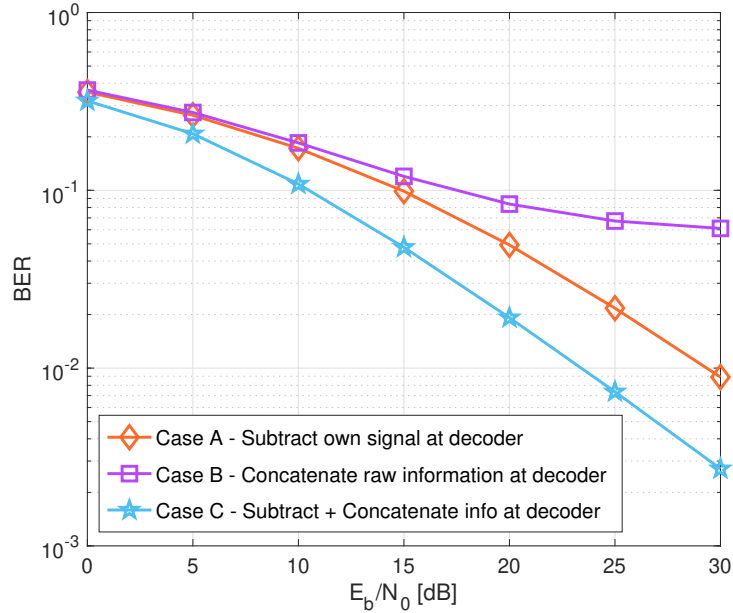


Figure 5.4.3: Comparison of proposed AE framework with different Lambda layers (Cases A–C) in the NN decoder of the terminal nodes.

merits of Remarks 5, 6.

AE-based BCM Design under RBF Channels

For the conventional scenario, we consider QPSK modulation with $(7, 4)$ Hamming coding. We consider $(n, k) = (7, 8)$ under Rayleigh block fading (RBF) channels such that channel remains constant for $n = 7$ and then changes randomly. We analyze the three possible ways of designing the Lambda layers in the NN decoder of the AE framework below:

- *Case A: Subtract own signal at the decoder* – This scenario is inspired by the process of decoding used in the conventional method, where a terminal node first subtract its own transmitted signal and then perform channel equalization. For this scenario, we propose to pre-process the received signal at the NN-based decoder of each Γ^{th} terminal node's decoder using two Lambda layers, given by $\mathbf{L}_L(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) = \mathbf{L}_5(\mathbf{L}_4(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma))$, detailed as follows:

L4: Each Γ^{th} terminal node's decoder removes its own signal from the received signal \mathbf{y}_Γ as $\bar{\mathbf{y}}_\Gamma = \mathbf{y}_\Gamma - \alpha h_{\Gamma r} h_{r\Gamma} \mathbf{x}_\Gamma$.

L5: Each Γ^{th} terminal node's decoder performs two-step channel equalization on $\bar{\mathbf{y}}_\Gamma$ by utilizing first-hop and second-hop channels of the intended signal, i.e. $h_{\Gamma r}$ and $h_{r\Gamma}$.

- *Case B: Concatenate all information at the decoder* – This scenario is inspired

Table 5.4.1: Insights for AE-based BCM design for the TW-AF relay networks.

Model	Minimum Euclidean distance $d_{E_{\min}}$ (\uparrow better)		Normalized second moment E_n (\downarrow better)		Normalized fourth moment χ		Constellation figure of merit CFM (\uparrow better)	
	A	B	A	B	A	B	A	B
	<i>Conventional QPSK + Hamming Code</i>	1.41	1.41	3.5	3.5	1	1	2
<i>AE – Case A</i>	0.3	0.3	76	76	1.004	1.004	0.1	0.1
<i>AE – Case B</i>	0.2	0.05	145	2847	1	1	0.05	0.003
<i>AE – Case C</i>	1.62	1.61	2.7	2.7	1	1	2.6	2.6
<i>AE proposed in [57]</i>	0.94	1.39	7.83	3.62	1	1	0.9	1.9

by the general thinking that NN should be given all the raw information and allowed to learn on its own. For this scenario, we propose to pre-process the received signal at the NN-based decoder of each Γ^{th} using two Lambda layers, given by $\mathbf{L}_L \left(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma \right) = \mathbf{L7} \left(\mathbf{L6} \left(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma \right) \right)$, detailed as follows:

L6: Each Γ^{th} terminal node’s decoder has a *Concatenate* Layer, which concatenates all the raw information – (i) received signal at Γ^{th} terminal node’s decoder, (ii) own transmitted signal by Γ^{th} terminal node’s encoder, (iii) amplification factor, and (iv) all the necessary channel information $(h_{\Gamma_r}, h_{r\Gamma}, h_{\Gamma_r})$.

L7: The output of the concatenated layer is passed through a *Batch-normalization* Layer \mathcal{B}_N .

- *Case C: Subtract own signal + Concatenate all the information at the decoder* – This scenario combines the Lambda layers in *Case A* and *Case B* in order. For this scenario, we propose to pre-process the received signal at the NN-based decoder of each Γ^{th} terminal node’s decoder using four Lambda layers (defined above), given by $\mathbf{L}_L \left(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma \right) = \mathbf{L7} \left(\mathbf{L6} \left(\mathbf{L5} \left(\mathbf{L4} \left(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma \right) \right) \right) \right)$.

The processed output of Lambda layers \mathbf{L}_L is given to the NN decoder in each terminal to predict the output bits \mathbf{u}_Γ at the Γ^{th} terminal node.

Now, we design three AE frameworks utilizing the three cases *Cases A–C*, and evaluate the BER versus transmit SNR (E_b/N_0) performance in Fig. 5.4.3. For greater insights into the optimality of designed BCM in $2n$ -dimensional space, we propose to employ various metrics such as, minimum Euclidean distance $d_{E_{\min}}$ (see Definition 16), normalized second-order moment E_n (see Definition 17), normalized fourth-order moment or Kurtosis χ (see Definition 18), and constellation figure of merit (CFM) (see Definition 20), detailed in Table 5.4.1. In Fig. 5.4.3, *Case B* performs the worst because the proposed NN decoders are not designed to handle complex numbers; thus, by giving all the raw information, the NN decoder is not able to infer that its own signal needs to be subtracted

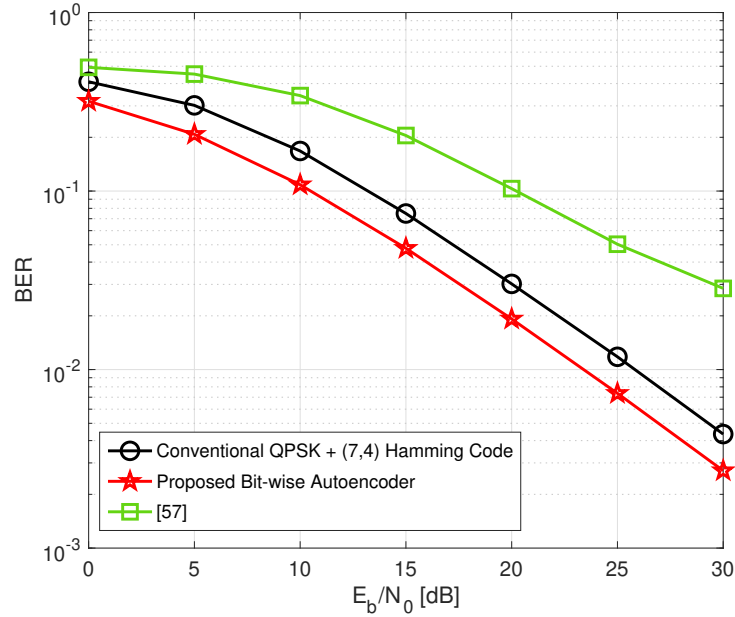


Figure 5.4.4: Proposed BWAEE v/s Conventional v/s [57].

and channel equalization needs to be performed. This problem becomes even more visible at higher SNR regimes, where noise is minimal. For the *Case A* where the decoder subtracts its own signal and performs channel equalization, BER reduces with increasing SNR. Suggesting that until complex-valued NN [83] is utilized, it is best to perform all the complex domain mathematical operations in Lambda layers. The *Case C* performs the best because we first perform all the complex domain mathematical operations in the Lambda layer followed by concatenating all the raw information. In Table 5.4.1, normalized fourth moment $\chi = 1$ in all three cases indicating always spherical codes are formed in $2n$ -dimensional space.

Now, we compare the transmit SNR versus BER performance in Fig. 5.4.4. In the proposed bit-wise AE framework, we keep the Lambda layers according to *Case C* (due to the reasons detailed above). We can see that the proposed bit-wise AE framework outperform the symbol-wise AE proposed in [57] by 11 dB and conventional QPSK modulation with (7, 4) Hamming code by 3 dB. This is because (as seen in Table 5.4.1) the BCM designed by the proposed AE has a minimum Euclidean distance for terminal nodes as $\{1.62, 1.61\}$, while that for the symbol-wise AE in [57] remains $\{0.94, 1.39\}$ and $\{1.41, 1.41\}$ in the conventional QPSK modulation. As the minimum Euclidean distance is same for both terminal nodes in the proposed bit-wise AE framework, thus it is also ensuring that both the terminal nodes have the similar QoS, while we cannot see it in the symbol-wise AE framework. Further, the major reason for such performance gains over [57] is because of two reasons – (1) we employ a bit-wise AE that performs automatic bit-labeling possibly in a Gray coded format, while bit labeling in the symbol-wise AE in [57] needs to be performed heuristically solving a $2^k!$ combinatorial problem, and (2) we utilize a

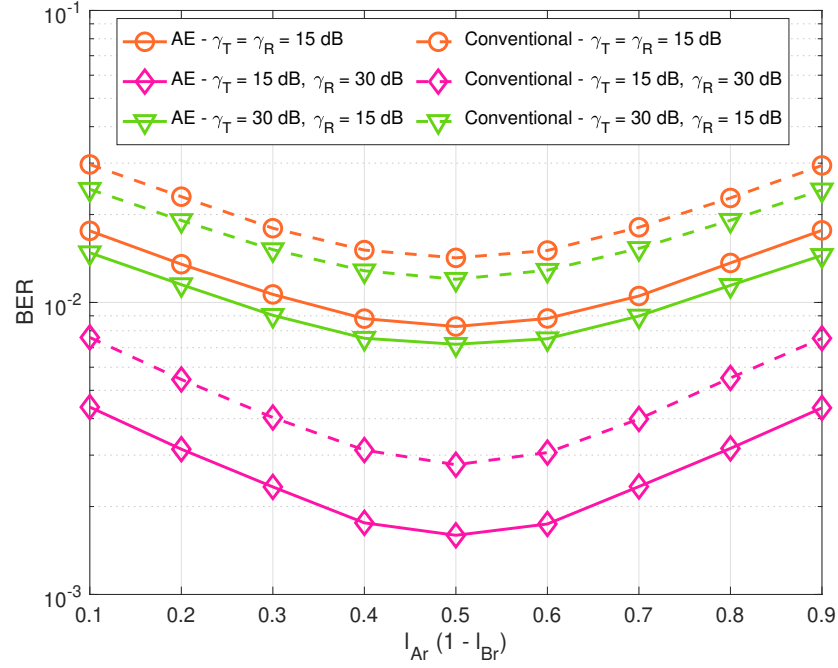


Figure 5.4.5: Optimal TW-AF relay location for proposed AE framework – Small-scale fading + Path-loss (No shadow fading).

conventional TW-AF relay node that simply amplifies and re-transmit the signal, over the NN-based relay node in [57].

Optimal TW-AF Relay Location for Proposed AE Framework

Now, we determine the optimal TW-AF relay location for the proposed AE framework. Let the distance between terminal A, B to relay node be given by l_{Ar}, l_{Br} . Let $\vartheta_{Ar}, \vartheta_{Br}$ capture the shadow fading in the $A \Leftrightarrow R$ and $B \Leftrightarrow R$ transmissions, assumed to be log-normal random variables with zero mean and standard deviation σ_{Ar} and σ_{Br} , respectively [84], [85]. Let β denotes the path-loss exponent [84]. Then, the signals received at the relay node and destination node are given as:

$$y_r = \sqrt{P_A \vartheta_{Ar} l_{Ar}^{-\beta}} h_{Ar} x_A + \sqrt{P_B \vartheta_{Br} l_{Br}^{-\beta}} h_{Br} x_B + n_r \quad (5.17)$$

$$y_\Gamma = \sqrt{P_r \vartheta_{\Gamma r} l_{\Gamma r}^{-\beta}} h_{r\Gamma} \alpha y_r + n_\Gamma \quad (5.18)$$

$$\bar{y}_\Gamma = y_\Gamma - \sqrt{P_\Gamma \vartheta_{\Gamma r} l_{\Gamma r}^{-\beta}} h_{r\Gamma} \alpha h_{\Gamma r} x_\Gamma \quad (5.19)$$

$$\alpha = \left(P_A \vartheta_{Ar} l_{Ar}^{-\beta} |h_{Ar}|^2 + P_B \vartheta_{Br} l_{Br}^{-\beta} |h_{Br}|^2 + \sigma_r^2 \right)^{-1/2} \quad (5.20)$$

We utilize the AE framework with *Case C* Lambda layers and keep $\beta = 3$. For the sake of simplicity, we keep the reference distance between the terminal nodes as $l_{Ar} + l_{Br} = 1$. In Fig. 5.4.5, we vary $l_{\Gamma r} = \{0.1, 0.2, \dots, 0.9\}$ to train and test a AE for each distance.

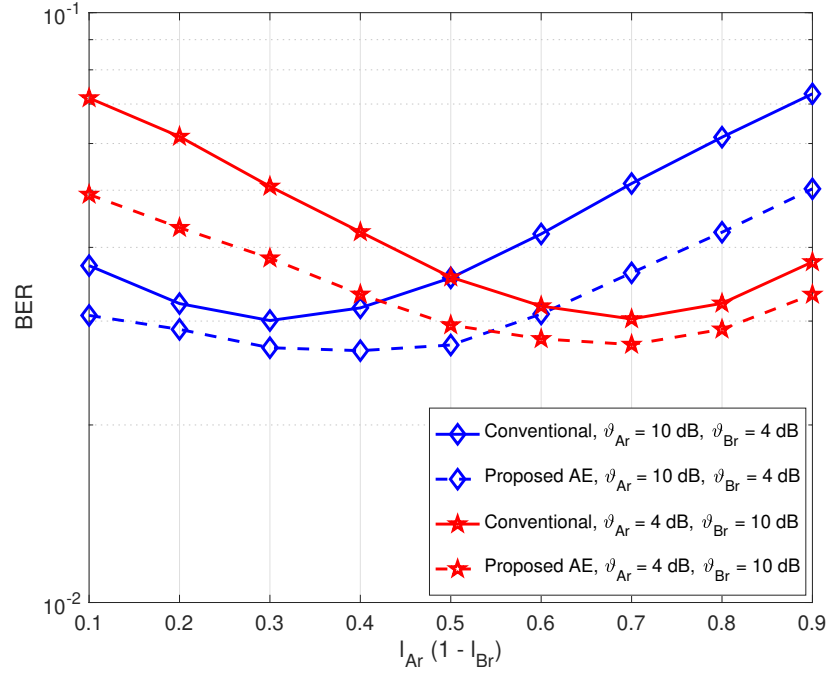


Figure 5.4.6: Optimal TW-AF relay location for proposed AE framework – Small-scale fading + Path-loss + Shadow fading.

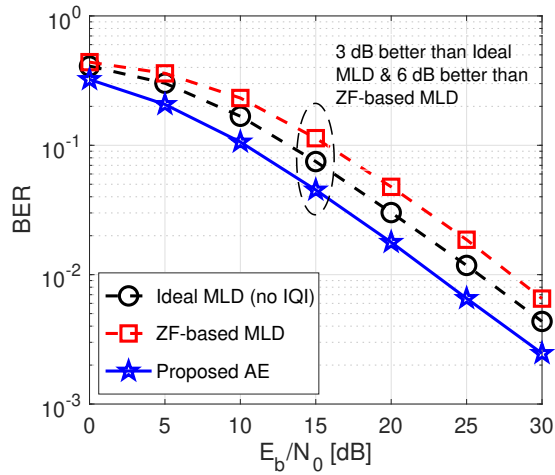
Throughout this section, we denote the transmit SNR of both the terminal node as γ_T in the first phase transmission and transmit SNR of the relay node as γ_R in the second phase transmission. Our proposed AE outperforms the conventional method for all the relay locations. We show optimal distance by d_{opt} .

In Fig. 5.4.5, we consider no shadow fading in the wireless links, and then the optimal relay location is always in the center, i.e., $d_{\text{opt}} = 0.5$ because the terminal nodes have a distance of l_{Ar} and $(1 - l_{Ar})$ from the relay node, and same transmit SNRs, thus if we keep the TW-AF relay node closer to terminal A , the signal of the terminal B deteriorates and vice-versa.

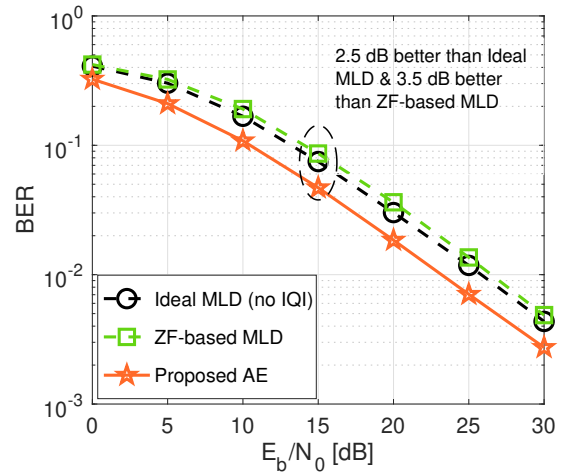
In Fig. 5.4.6, we vary shadow fading with fixed transmit SNR $\gamma_T = \gamma_R = 15$ dB. Naturally, when the link quality between a terminal node and a TW-AF relay node is worse, then the optimal relay location is closer to that terminal node. Thus, we see that when $v_{Ar} > v_{Br}$ then the optimal TW-AF relay location is $d_{\text{opt}} = 0.3$ (closer to terminal A) and vice-versa.

5.4.2 TW-AF Relay Networks with I/Q Imbalance

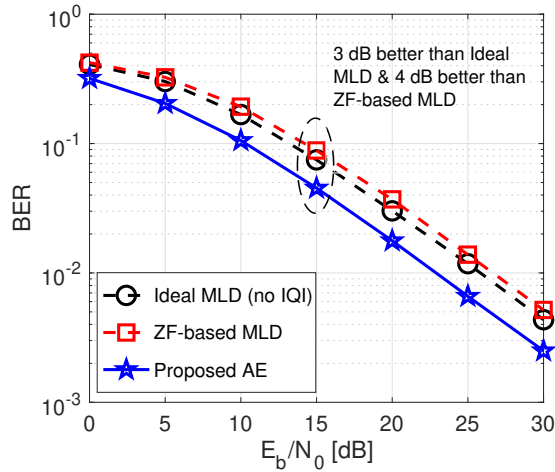
In this subsection, we evaluate the performance of AE-based BCM design for the TW-AF relay networks with I/Q imbalance. For the conventional scenario, we consider QPSK modulation with $(7, 4)$ Hamming coding and perform ZF-based IQI compensation with MLD decoding (ZF-based MLD). Also, we evaluate the conventional TW-AF relay net-



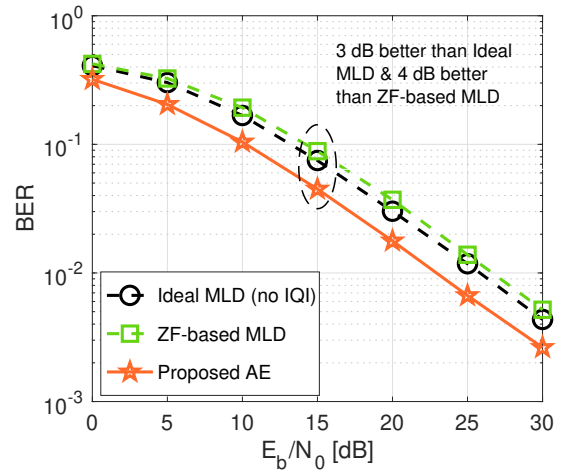
(a) $\xi = 0.6$ for fixed phase offset $\phi = 10^\circ$.



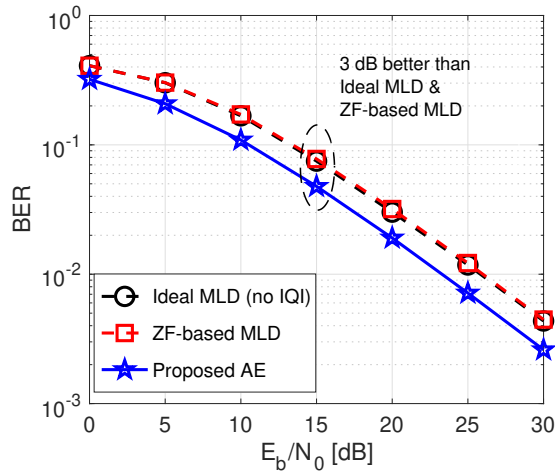
(b) $\phi = 0^\circ$ for fixed amplitude offset $\xi = 0.8$.



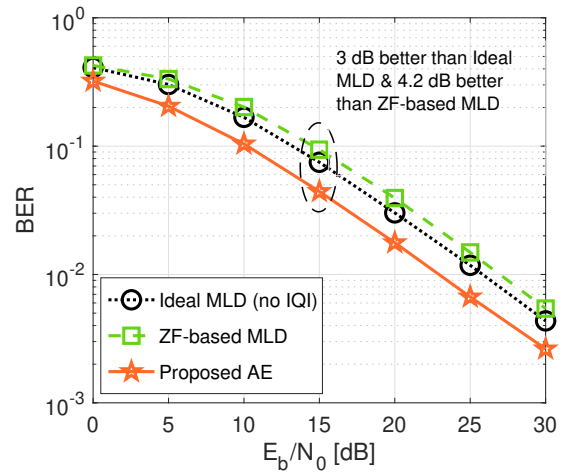
(c) $\xi = 0.8$ for fixed phase offset $\phi = 10^\circ$.



(d) $\phi = 10^\circ$ for fixed amplitude offset $\xi = 0.8$.



(e) $\xi = 1$ for fixed phase offset $\phi = 10^\circ$.



(f) $\phi = 20^\circ$ for fixed amplitude offset $\xi = 0.8$.

Figure 5.4.7: BER performance for TW-AF relay networks with I/Q-Imbalance.

work with no IQI, i.e., $\xi = 1, \phi = 0^\circ$ with MLD and label it as ‘MLD (no IQI).’ We consider $(n, k) = (7, 8)$ under RBF channels such that the channel remains constant for $n = 7$ and then changes randomly. For sake of simplicity, we assume $\xi_T = \xi_R = \xi$ and $\phi_T = \phi_R = \phi$ throughout this section.

Furthermore, in Section 5.4.1, we see that performing all complex domain operations in Lambda layer followed by concatenating raw information is the best approach to maximize the decoding performance (in terms of BER performance) of the NN decoders at each terminal node. Thus, for the TW-AF relay network with IQI, we propose to pre-process the received signal at the NN-based decoder using four Lambda layers, detailed as: $\mathbf{L}_L \left(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma \right) = \mathbf{L}_{11} \left(\mathbf{L}_{10} \left(\mathbf{L}_9 \left(\mathbf{L}_8 \left(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma \right) \right) \right) \right)$, as:

- L8:** Each Γ^{th} terminal node’s decoder removes its own signal and SI from own signal from the received signal \mathbf{y}_Γ^{IQ} similar to (5.6).
- L9:** Each Γ^{th} terminal node’s decoder multiplies the obtained signal with \mathbf{A}^{-1} described in (5.8) to obtain $\hat{\mathbf{y}}_\Gamma^{IQ}$ similar to (5.9).
- L10:** Each Γ^{th} terminal node’s decoder has a *Concatenate* Layer $\mathcal{C}_C(\cdot)$, which concatenates all the raw information – (i) received signal at Γ^{th} terminal node’s decoder, (ii) own transmitted signal by Γ^{th} terminal node’s encoder, (iii) amplification factor, (iv) all the necessary channel information $(h_{\Gamma r}, h_{r\Gamma}, h_{\Upsilon r})$, (v) $\Lambda_{\Upsilon\Gamma}$ and $\Omega_{\Upsilon\Gamma}$ (defined in (5.6)), and (vi) $(K_1 G_1 h_{\Gamma r} h_{r\Gamma} + K_2 G_2 h_{\Gamma r}^* h_{r\Gamma}^*)$, $(K_1 G_2^* h_{\Gamma r} h_{r\Gamma} + K_2 G_1^* h_{\Gamma r}^* h_{r\Gamma}^*)$
- L11:** The output of the concatenated layer is passed through a batch-normalization layer \mathcal{B}_N .

In particular, **L8, L9** helps to perform complex-domain operations on the received signals (SIC and ZF-based IQI compensation), and **L10, L11** helps to concatenate all the raw information as real values so that the dense layers in the NN decoder can decode the signal efficiently, by forming valuable representations.

In Fig. 5.4.7, we analyze the BER performance by varying the amplitude offset (in Fig. 5.4.7a, 5.4.7c, 5.4.7e) and phase offset (in Fig. 5.4.7b, 5.4.7d, 5.4.7f). Clearly, the proposed AE-based BCM designs outperform the TW-AF relay network with ZF-based MLD. Moreover, the proposed AE-based BCM designs remains unaffected of the phase and amplitude offsets, thereby outperforming MLD with no IQI. The aforementioned gains can be attributed to the fact that we are performing BCM designs in $2n$ -dimensional space, leading to the design of 2^k codewords as a spherical code in a Gray-coded format. This is possibly leading to the maximization of the minimum Euclidean distance and minimum Hamming distance between the codewords. But also, because of the pre-processing of the received signals using the four lambda layers at the NN decoders, the terminal nodes are removing the SI from their own signals, minimizing the effects of IQI

and concatenating all raw information to make NN decoder learn good representations in $2n$ -dimensional space to decode the signal efficiently.

In Fig. 5.4.7, we can see that as the amplitude or phase offset increases, the BER performance gains of the proposed AE-based BCM design increase in comparison to ZF-based MLD and remains constant in contrast to MLD with no IQI. This is because the AE-based BCM design can altogether remove the deteriorating impacts of IQI, whereas the ZF-based MLD's performance deteriorates with increasing IQI.

5.5 Conclusion

In this chapter, we proposed an AE-based BCM design for the TW-AF relay network impacted by IQI. We also propose to utilize a conventional TW-AF relay node. The proposed AE framework outperform the BER performance in [57] by 11 dB and ASR performance in [38] by 3.5 dB. These gains are analyzed using various coded-modulation metrics, similar to Chapter 3. Further, we show that until complex-valued NN [83] are utilized, it is best to perform complex domain operations in Lambda layers and concatenate raw information together. This also affirms the benefits of employing ZF-based SIC in Chapter 4. Also, by determining estimated MI, we show that (1) AE is well approximated at moderate SNR values, similar to Chapter 3, and (2) proposed bit-wise AE is ensuring similar QoS by designing similar BCM designs for both the terminal nodes, unlike the symbol-wise AE, showing an additional advantage of a bit-wise AE over symbol-wise AE framework.

Also, we show that the optimal relay location in an AE framework is in the center in case of no shadow fading and close to the terminal node with higher shadow fading.

Furthermore, we show that the proposed AE-based BCM designs for the TW-AF relay network can altogether remove the deteriorating impacts of IQI at the terminal nodes, outperforming the MLD with no IQI by 3 dB. This shows that the BCM designs are even able to remove the residual errors after the ZF-based removal of IQI. We focus on two-way DF relaying networks with hardware impairments in Chapter 7.

Appendix

5.A Derivation of Binary CE Loss and Estimated MI for TW-AF Relay networks with IQI

As shown in (5.14), we train the proposed AE-based framework by minimizing the total binary cross-entropy (CE) loss for both the terminal nodes $\Gamma \rightarrow R \rightarrow \Upsilon$ and $\Upsilon \rightarrow R \rightarrow \Gamma$ together, where $\{\Gamma, \Upsilon\} = \{A, B\}$ and $\Gamma \neq \Upsilon$. For any received block of complex baseband symbols, \mathbf{y}_Γ , the total binary CE loss in (5.14) can also be represented as

$$\mathcal{J}(\boldsymbol{\theta}_{A_{Tx}}, \boldsymbol{\theta}_{B_{Tx}}, \boldsymbol{\theta}_{A_{Rx}}, \boldsymbol{\theta}_{B_{Rx}}) = \frac{1}{2} \sum_{\substack{\{\Gamma, \Upsilon\} = \{A, B\}, \\ \Gamma \neq \Upsilon}} H\left(p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}), \tilde{p}_{\boldsymbol{\theta}_{\Gamma_{Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)\right) \quad (5.21)$$

where $H\left(p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}), \tilde{p}_{\boldsymbol{\theta}_{\Gamma_{Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)\right)$ denotes the binary CE loss for the transmission between $\Upsilon \rightarrow R \rightarrow \Gamma$. Specifically, it denotes the binary CE loss between the true distribution $p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ})$ at the NN encoder of the Υ^{th} terminal node and the learnt distribution $\tilde{p}_{\boldsymbol{\theta}_{\Gamma_{Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)$ at the NN decoder of the Γ^{th} terminal node. Please note all the parameters $\boldsymbol{\theta}_{\Gamma_{Tx}}$ and $\boldsymbol{\theta}_{\Upsilon_{Rx}}$, where $\{\Gamma, \Upsilon\} = \{A, B\}$ and $\Gamma \neq \Upsilon$ are learned “together”. Now, we firstly derive the binary CE loss by taking the expectation of (5.21) with

respect to \mathbf{y}_Γ^{IQ} and \mathbf{x}_Γ , as follows

$$\begin{aligned}
 & \mathcal{J}(\boldsymbol{\theta}_{A_{Tx}}, \boldsymbol{\theta}_{B_{Tx}}, \boldsymbol{\theta}_{A_{Rx}}, \boldsymbol{\theta}_{B_{Rx}}) \\
 & \stackrel{(a)}{=} \frac{1}{2} \sum_{\substack{\{\Gamma, \Upsilon\}=\{A,B\}, \\ \Gamma \neq \Upsilon}} \mathbb{E}_{\mathbf{y}_\Gamma, \mathbf{x}_\Gamma} \left[H \left(p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}), \tilde{p}_{\boldsymbol{\theta}_{\Upsilon_{Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \right) \right] \\
 & \stackrel{(b)}{=} \frac{1}{2} \sum_{\substack{\{\Gamma, \Upsilon\}=\{A,B\}, \\ \Gamma \neq \Upsilon}} \sum_{m=1}^k \mathbb{E}_{\mathbf{y}_\Gamma, \mathbf{x}_\Gamma} \left[H \left(p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}), \tilde{p}_{\boldsymbol{\theta}_{\Upsilon_{Rx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \right) \right] \\
 & \stackrel{(c)}{=} \frac{-1}{2} \sum_{\substack{\{\Gamma, \Upsilon\}=\{A,B\}, \\ \Gamma \neq \Upsilon}} \sum_{m=1}^k \mathbb{E}_{\mathbf{y}_\Gamma, \mathbf{x}_\Gamma} \left[p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}) \log \tilde{p}_{\boldsymbol{\theta}_{\Upsilon_{Rx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \right] \\
 & \stackrel{(d)}{=} \frac{-1}{2} \sum_{\substack{\{\Gamma, \Upsilon\}=\{A,B\}, \\ \Gamma \neq \Upsilon}} \sum_{m=1}^k \int_{\mathbf{y}_\Gamma} \int_{\mathbf{x}_\Gamma} p(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}) \log \tilde{p}_{\boldsymbol{\theta}_{\Upsilon_{Rx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) d\mathbf{y}_\Gamma d\mathbf{x}_\Gamma \\
 & \stackrel{(e)}{=} \frac{-1}{2} \sum_{\substack{\{\Gamma, \Upsilon\}=\{A,B\}, \\ \Gamma \neq \Upsilon}} \sum_{m=1}^k \int_{\mathbf{y}_\Gamma} \int_{\mathbf{x}_\Gamma} p(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}) \\
 & \quad \log \left[\frac{p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}) \tilde{p}_{\boldsymbol{\theta}_{\Upsilon_{Rx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)}{p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ})} \right] d\mathbf{y}_\Gamma d\mathbf{x}_\Gamma \\
 & \stackrel{(f)}{=} \frac{1}{2} \sum_{\substack{\{\Gamma, \Upsilon\}=\{A,B\}, \\ \Gamma \neq \Upsilon}} \sum_{m=1}^k \int_{\mathbf{y}_\Gamma} \int_{\mathbf{x}_\Gamma} p(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}) \log \left[\frac{p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ})}{\tilde{p}_{\boldsymbol{\theta}_{\Upsilon_{Rx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)} \right] d\mathbf{y}_\Gamma d\mathbf{x}_\Gamma \\
 & \quad - \frac{1}{2} \sum_{\substack{\{\Gamma, \Upsilon\}=\{A,B\}, \\ \Gamma \neq \Upsilon}} \sum_{m=1}^k \int_{\mathbf{y}_\Gamma} \int_{\mathbf{x}_\Gamma} p(\mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}) \log \left[p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(u_\Upsilon^m | \mathbf{y}_\Gamma^{IQ}) \right] d\mathbf{y}_\Gamma d\mathbf{x}_\Gamma \\
 & \stackrel{(g)}{=} \frac{1}{2} \sum_{\substack{\{\Gamma, \Upsilon\}=\{A,B\}, \\ \Gamma \neq \Upsilon}} D_{KL} \left(p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}) \parallel \tilde{p}_{\boldsymbol{\theta}_{\Upsilon_{Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \right) + H_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}) \\
 & \stackrel{(h)}{=} \frac{1}{2} \sum_{\substack{\{\Gamma, \Upsilon\}=\{A,B\}, \\ \Gamma \neq \Upsilon}} D_{KL} \left(p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}) \parallel \tilde{p}_{\boldsymbol{\theta}_{\Upsilon_{Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \right) + H(\mathbf{u}_\Upsilon) - I_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon; \mathbf{y}_\Gamma^{IQ})
 \end{aligned} \tag{5.22}$$

where $D_{KL} \left(p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}) \parallel \tilde{p}_{\boldsymbol{\theta}_{\Upsilon_{Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \right)$ denotes the Kullback-Leibler (KL)-divergence loss between the true distribution $p_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ})$ at the NN encoder of the Υ^{th} terminal node and the learnt distribution $\tilde{p}_{\boldsymbol{\theta}_{\Upsilon_{Rx}}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma)$ at the NN decoder of the Γ^{th} terminal node. Please note all the parameters $\boldsymbol{\theta}_{\Upsilon_{Tx}}$ and $\boldsymbol{\theta}_{\Upsilon_{Rx}}$, where $\{\Gamma, \Upsilon\} = \{A, B\}$ and $\Gamma \neq \Upsilon$ are learned “together”. $H(\mathbf{u}_\Upsilon)$ denotes entropy of the input bits \mathbf{u}_Υ at the Υ^{th} terminal node, and $I_{\boldsymbol{\theta}_{\Upsilon_{Tx}}}(\mathbf{u}_\Upsilon; \mathbf{y}_\Gamma^{IQ})$ is the MI between the input bits \mathbf{u}_Υ and the received

signal at the Γ^{th} terminal node \mathbf{y}_Γ^{IQ} with parameters $\boldsymbol{\theta}_{\Gamma T_x}$.

Moreover, the (5.22) is derived using the following steps: (a), (b) is obtained by the taking expectation over \mathbf{y}_Γ^{IQ} and \mathbf{x}_Γ , (c) comes from the definition of binary CE loss, in (d) we open the expectation, in (e) we multiply and divide by $p_{\boldsymbol{\theta}_{\Gamma T_x}}(u_\Gamma^m | \mathbf{y}_\Gamma^{IQ})$, in (f) we open the log function, (g) comes from the definition of KL divergence loss and conditional entropy, and (h) utilizes the identity $H(a|b) = H(a) - I(a; b)$ [49].

The estimated MI is defined in Definition 14. Thus, for TW-AF the estimated MI (\mathcal{I}) can be given as follows

$$\mathcal{I} := \frac{1}{2} \sum_{\substack{\{\Gamma, \Upsilon\}=\{A, B\}, \\ \Gamma \neq \Upsilon}} I_{\boldsymbol{\theta}_{\Gamma T_x}}(\mathbf{u}_\Upsilon; \mathbf{y}_\Gamma^{IQ}) - D_{KL} \left(p_{\boldsymbol{\theta}_{\Gamma T_x}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}) || \tilde{p}_{\boldsymbol{\theta}_{\Gamma R_x}}(\mathbf{u}_\Upsilon | \mathbf{y}_\Gamma^{IQ}, \mathbf{x}_\Gamma) \right) \quad (5.23)$$

Now, using (5.22), the estimated MI in (5.23) becomes

$$\mathcal{I} := \frac{H(\mathbf{u}_A) + H(\mathbf{u}_B)}{2} - \mathcal{J}(\boldsymbol{\theta}_{A T_x}, \boldsymbol{\theta}_{B T_x}, \boldsymbol{\theta}_{A R_x}, \boldsymbol{\theta}_{B R_x}) \quad (5.24)$$

Since the first term on R.H.S, $\frac{H(\mathbf{u}_A) + H(\mathbf{u}_B)}{2}$ in (5.24), remains a constant, thus the changes in the estimated MI in (5.23) only depends on the total binary CE loss term $\mathcal{J}(\boldsymbol{\theta}_{A T_x}, \boldsymbol{\theta}_{B T_x}, \boldsymbol{\theta}_{A R_x}, \boldsymbol{\theta}_{B R_x})$.

Chapter 6

AE-based Full Duplex

Amplify-and-Forward Relay Networks

6.1 Introduction

As seen in previous chapters 2–5, the autoencoder (AE)-based wireless communication networks in half-duplex (HD) transmission mode has appeared as a promising solution to achieve significant bit-error-rate (BER) performance gains for a rate $R = k/n$ [bits/channel-reuse]. However, all of these chapters and previous works [12]–[40] focus on designing AE-based block coded modulation (BCM) and differential BCM (d-BCM) designs for HD transmission mode.

Recently, the full-duplex (FD) mode is recognized as an enabling technology to realize the expected gains in the future networks, as it can double the spectral efficiency by establishing concurrent transmission and reception on the same temporal and spectral resources [24, 80, 86, 87, 88, 89, 90, 91, 92]. However, the self-interference (SI) leaking from the signal transmitted by the relay node interferes with the signal received at the relay node, limiting spectral efficiency gains. Recently, the superior self-interference cancellation (SIC) methods have attested to the facilitation of FD transmission [24, 80, 86, 87, 88, 89, 90, 91, 92, 93]. In particular, the SIC can be done efficiently by employing multiple techniques, such as antenna isolation, which eliminates the self-interference from -50 dB to -100 dB, analog-domain suppression, which reduces the self-interference by about -50 dB, and digital-domain suppression, which reduces the self-interference almost -60 dB [94]. However, even with multiple SIC techniques, a residual self-interference (RSI) is always present in the system, deteriorating the FD performance. Thus, tackling the RSI is pivotal to realize the double spectral efficiency promised in a FD mode.

The analysis and optimization of FD amplify-and-forward (FD-AF) relaying in the presence of RSI has been studied widely [95, 96, 97], wherein, the relay's function is to multiply the received signal in the presence of the RSI with a variable gain, also referred

to as amplification factor, and re-transmit the signal to the destination node. The relays may need full or partial channel state information (CSI) of the source to relay link and channel gains of the SI channel for determining the amplification factor. Besides, the destination node would need the CSI of the overall channel of source-relay-destination and sometimes the channel gains of the SI channel. To avoid channel estimation at the relay and destination nodes of an AF relay network, operating in HD mode, the differential modulation techniques are usually applied while utilizing only the second-order statistics of the channels at the relays for signal amplification [62], [63]. Designing differential modulations for FD mode still remains an open research problem.

In this chapter, for the first time, we propose an AE-based framework for FD-AF relay network with RSI at the relay node. We now summarize the major contributions of this chapter below.

- We propose a bit-wise AE-based FD-AF relay network, where we consider NN-based encoder-decoder at the source and destination nodes, and the conventional FD-AF relay node¹ operating in the presence of RSI. Depending on the availability of the CSI knowledge, we consider three scenarios. Firstly, we propose a BCM design for the FD-AF relay networks in the presence of perfect CSI knowledge. Secondly, we also analyze the proposed BCM design for the FD-AF relay networks with imperfect CSI knowledge. Thirdly, we completely remove the necessity of CSI knowledge by proposing differential FD-AF relay networks – (i) we design the amplification factor for conventional FD-AF relay node by including the second-order channel statistics of the RSI, and (ii) we propose the d-BCM design.
- Focussing on the convergence of the training of the proposed AE framework – (i) we analyze the convergence of the AE frameworks by estimating the MI between the input-output of the AE for varying RSI and signal-to-noise-ratio (SNR) levels, and (ii) we provide the necessary conditions for AE's convergence by showing that once the validation loss has converged and the NN encoder of AE designs 2^k codewords during the training, the AE has converged to its maximum potential of decoding the signal.
- With the aim to open the black-box of the AE-based BCM design, we reveal the five distinct observations of 2^k codewords designed in $2n$ -dimensional space by the proposed end-to-end training of the AE frameworks. Specifically, we utilize the minimum Euclidean distance, packing density, average Hamming distance, and Kurtosis to understand the designed codewords' observations at the source node's NN encoder.

¹We do not employ any neural network at the FD-AF relay node.

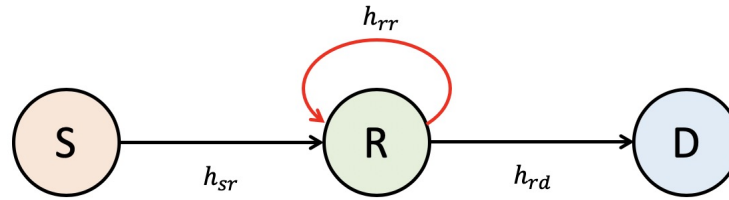


Figure 6.1.1: System model for full-duplex amplify-and-forward relay networks.

- We design a single Fixed AE framework that can efficiently handle varying rates. We train the AE framework to remain highly generalizable of the testing signal-to-noise-ratio (SNR) or RSI levels. Lastly, we also show that the proposed AE frameworks are highly reproducible even with different training samples and weight initialization.

6.2 System Model

We consider an FD-AF relay network as shown in Fig. 6.1.1, consisting of a source node (S) that wants to transmit its signal to the destination node (D), with the aid of an FD-AF relay node (R). Each of the source and destination nodes has a single antenna for transmission and reception, respectively. The relay node has two antennas, one for the reception and the other for transmission. We assume that the direct link between the source and destination node is strongly attenuated because of severe path-loss and shadowing effects.

6.2.1 Signal Transmission – Reception Process

In this chapter, we assume K blocks of symbols, each of length n , is transmitted from source node to destination node with the aid of the FD-AF relay node in $\kappa = \{1, \dots, K(n+1)\}$ time-instants. Specifically, we consider block-by-block encoding and decoding of k bits at the source and destination nodes, respectively, where n denotes the block length, while the signal transmission takes place as symbol-by-symbol. Thus, the transmission rate becomes $R = k/n$ [bits/channel use], which is double the transmission rate compared to HD AF relay networks (in Chapter 3, 4). In each time-instant (κ), the source node transmits the intended signal to the FD-AF relay node, which re-transmits the amplified signal (received at $(\kappa - 1)$ time-instant) to the destination node at the same time-instant (κ). We summarize the signal transmission and reception process in Table 6.2.1 for each $i^{\text{th}} \in \mathcal{U} := \{1, \dots, K\}$ block of data, where the relay node operates in HD mode for only the first and last time-instants and in FD mode otherwise.

Table 6.2.1: Process of signal transmission and reception of $i^{\text{th}} \in \mathcal{U}$ block of data.

Time-instants $[\kappa]$	[1]	[2]	[3]	\dots	[n]	[$n+1$]
Symbol Tx by S	$x_s[1]$	$x_s[2]$	$x_s[3]$	\dots	$x_s[n]$	–
Symbol Rx by R after SIC	$y_r[1]$	$y_r[2]$	$y_r[3]$	\dots	$y_r[n]$	–
Symbol Tx by R	–	$x_r[1]$	$x_r[2]$	\dots	$x_r[n-1]$	$x_r[n]$
Symbol Rx by D	–	$y_d[1]$	$y_d[2]$	\dots	$y_d[n-1]$	$y_d[n]$
Duplexing mode at R	HD	FD	FD	\dots	FD	HD

6.2.2 Modelling Residual Self Interference (RSI)

The RSI at the FD-AF relay node (R) can be modeled in two ways – (1) the complex Gaussian random model, where the RSI is modeled as the independent and identically distributed (i.i.d.) complex Gaussian random variables, having a similar effect as the noise and aims at emphasizing the effect of RSI on the performance [98], and (2) the general fading effect model, where the RSI is modeled as a statistical fading distribution, such as i.i.d. Rician/Rayleigh fading to model the RSI channel effectively [99]. In this chapter, we utilize the general fading effect model for RSI to characterize the RSI channel at the relay node R effectively. In particular, RSI is modeled by i.i.d. Rayleigh block-fading (RBF) channel $h_{rr} \sim \mathcal{CN}(0, \sigma_{rr}^2)$ [95], [96], such that it remains constant for n transmissions [97].

6.2.3 Signal Transmission Model and MLD Decoding

The source node (S) intends to transmit $\mathbf{u}_s \in \{0, 1\}^k$ bits, thus it first perform channel encoding $\bar{\mathbf{x}}_s = g_c(\mathbf{u}_s)$ to obtain $\{0, 1\}^j$ bits that are modulated to n complex baseband symbols $\mathbf{x}_s = g_m(\bar{\mathbf{x}}_s) \in \mathbb{C}^n$, such that $\|\mathbf{x}_s\|_2^2 = n$, where g_c and g_m denote the channel-coding and modulation functions. Then source node perform symbol-by-symbol transmission, and the signal received by the relay node after the SIC or under the presence of the RSI, at time-instant κ , is given by

$$y_r[\kappa] = \sqrt{P_s[\kappa]}h_{sr}[\kappa]x_s[\kappa] + \underbrace{h_{rr}[\kappa]x_r[\kappa]}_{\text{RSI}} + n_r[\kappa] \quad (6.1)$$

where P_s denotes the transmission power of source node, h_{sr} is the i.i.d. RBF channel with $h_{sr} \sim \mathcal{CN}(0, \sigma_{h_{sr}}^2 = 1)$, n_r is the additive white Gaussian noise (AWGN) at the relay node with $n_r \sim \mathcal{CN}(0, \sigma_r^2)$, and x_r is the amplified signal transmitted by the FD-AF relay node at the same time-instant κ , given by

$$x_r[\kappa] = \sqrt{P_r[\kappa]}\alpha[\kappa-1]y_r[\kappa-1] \quad (6.2)$$

where P_r denotes relay's transmission power and the amplification factor α is given as

$$\alpha[\kappa] = \left(P_s[\kappa]|h_{sr}[\kappa]|^2 + P_r[\kappa]|h_{rr}[\kappa]|^2 + \sigma_r^2 \right)^{-1/2} \quad (6.3)$$

Now, the signal received by the destination node (D) is represented as

$$\begin{aligned} y_d[\kappa] &= h_{rd}[\kappa]x_r[\kappa] + n_d[\kappa] = h_{rd}[\kappa]\sqrt{P_r[\kappa]}\alpha[\kappa-1]y_r[\kappa-1] + n_d[\kappa] \\ &= \underbrace{\sqrt{P_s[\kappa-1]P_r[\kappa]}\alpha[\kappa-1]h_{sr}[\kappa-1]h_{rd}[\kappa]x_s[\kappa-1]}_{\text{Desired Signal}} \\ &\quad + \underbrace{\sqrt{P_r[\kappa]}\alpha[\kappa-1]h_{rr}[\kappa-1]h_{rd}[\kappa]x_r[\kappa-1]}_{\text{RSI Signal}} \\ &\quad + \underbrace{\sqrt{P_r[\kappa]}\alpha[\kappa-1]h_{rd}[\kappa]n_r[\kappa-1] + n_d[\kappa]}_{\text{Noise}} \end{aligned} \quad (6.4)$$

where $h_{rd} \sim \mathcal{CN}(0, 1)$ is the i.i.d. RBF channel in second-hop and n_d is AWGN at the destination node with $n_d \sim \mathcal{CN}(0, \sigma_d^2)$. The destination node performs optimal maximum likelihood decoding (MLD), as follows:

$$\hat{x}_d = \arg \min_{x \in \mathcal{C}} \left\| y_d[\kappa] - \sqrt{P_s[\kappa-1]P_r[\kappa]}\alpha[\kappa-1]h_{sr}[\kappa-1]h_{rd}[\kappa]x \right\|^2 \quad (6.5)$$

where \mathcal{C} denotes all the possible alphabets. The decoder performs block-by-block channel-decoding using g_{cd} function, to obtain $\hat{\mathbf{u}}_s = g_{cd}(\hat{\mathbf{x}}_d)$, where $\hat{\mathbf{x}}_d \in \{0, 1\}^j$ and $\hat{\mathbf{u}}_s \in \{0, 1\}^k$.

6.2.4 Differential FD-AF Relay Networks - Without CSI

In the absence of the CSI knowledge, we propose to utilize traditional differential modulation and demodulation techniques at the source and destination nodes. For such scenarios, we propose to design the amplification factor for the FD-AF relay node by utilizing the second-order channel statistics of the first-hop channel between the source and relay node, and the RSI channel as

$$\alpha[\kappa] = \left(\sigma_{h_{sr}}^2 + \sigma_{rr}^2 + \sigma_r^2 \right)^{-1/2} \quad (6.6)$$

where the second-order channel statistics $\{\sigma_{h_{sr}}^2, \sigma_{rr}^2, \sigma_r^2\}$ can be obtained via long-term average of the received signals. Similar approximations have been employed for the HD-AF relays network in [62], [63]. To include the impact of RSI in FD scenarios we introduce the second-order channel statistics of the RSI channel in (6.6).

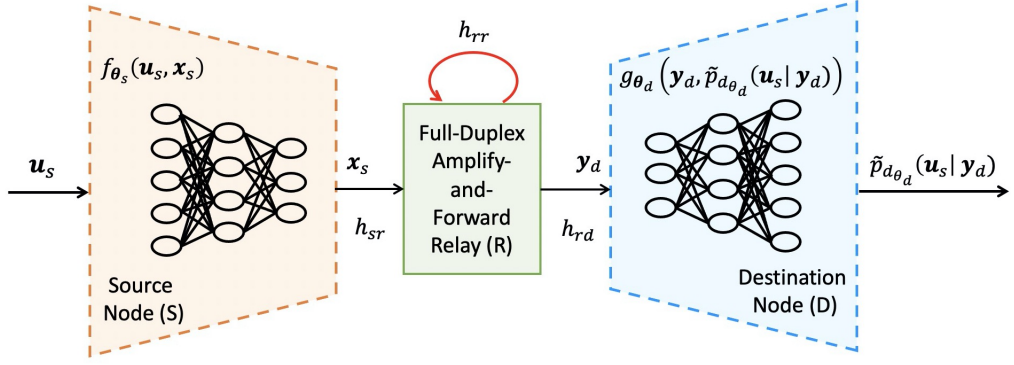
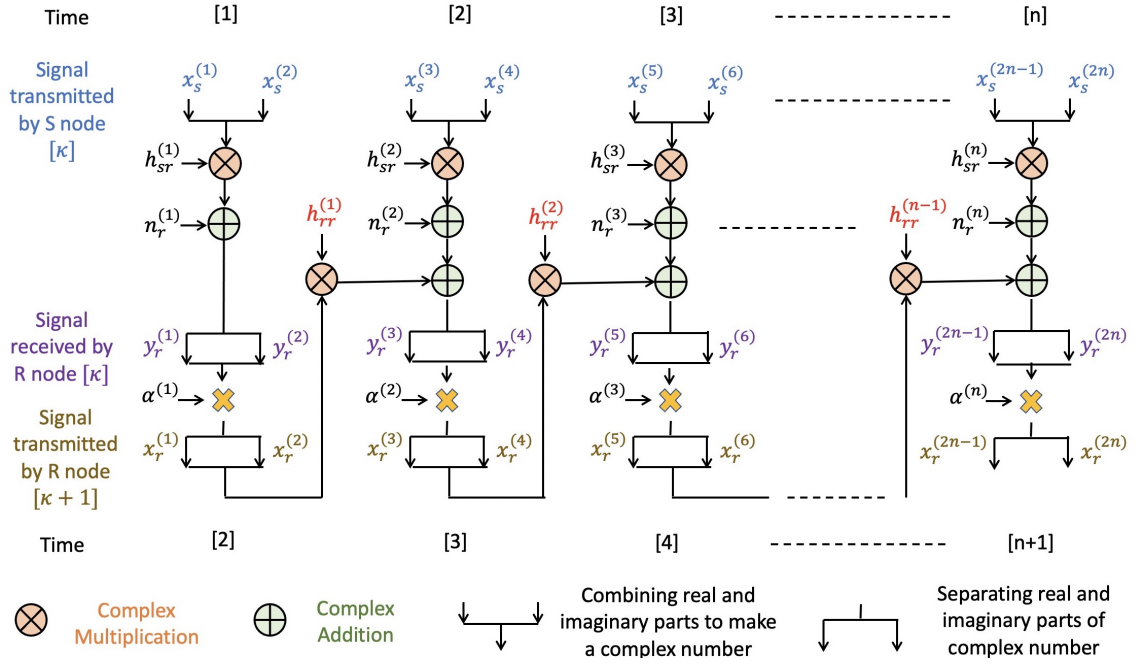


Figure 6.2.1: Proposed AE-based FD-AF relay networks.


 Figure 6.2.2: Signal transmission-reception of i^{th} block of data for the AE-based FD-AF relay network.

6.3 Proposed Autoencoder-based FD-AF Relay Networks

We propose a bit-wise fixed AE-based FD-AF relay networks as shown in Fig. 6.2.1, by considering NN-based source and destination nodes that performs joint BCM or d-BCM design and automatic bit-labelling of designed codewords. In this chapter, we utilize L fully-connected (dense) layers in the NN encoder-decoder pair of the AE framework, wherein the output of the $l^{\text{th}} \in \{1, \dots, L\}$ dense layer $\omega_l \in \mathbb{R}^{\delta_l}$ can be given as (2.2).

6.3.1 Designing of NN Encoder at the Source Node

The source node takes k bits $\mathbf{u}_s \in \{0, 1\}^k$ as input and maps it to n complex base-band symbols $\mathbf{x}_s \in \mathbb{C}^n$ (performing block-by-block encoding), by a mapping function $f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s)$, where θ_s is the NN encoder's parameters of the M dense layers, detailed in

Definition 6 and given as (2.3). We now perform symbol-by-symbol transmission and at any time-instant κ the symbol received by the FD-AF relay node, in the presence of RSI can be given as

$$y_r[\kappa] = h_{sr}[\kappa]f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s)[\kappa] + h_{rr}[\kappa]x_r[\kappa] + n_r[\kappa] \quad (6.7)$$

6.3.2 Designing of FD-AF Relay Node

Traditionally, the AF relaying scheme is designed to have minimal implementation complexity by receiving, amplifying, and re-transmitting the signal. Similar to Chapter 3 for HD-AF relay networks, we propose to utilize a conventional FD-AF relay node in this chapter. This is because NN-based processing at the FD-AF relay node, in the presence of RSI, worsens the decoding performance at the destination node (as detailed in Appendix 6.A). Thus, the signal transmitted by the FD-AF relay node becomes (6.2), where we utilize the amplification factor given in (6.3) and (6.6) for the BCM and d-BCM designs, respectively. Further, the process of signal-transmission reception remains the same as Section 6.2.1, and is summarized in Fig. 6.2.2.

6.3.3 Designing NN Decoder at the Destination Node

The signal received by the destination node is given as (6.4). Also, the designing of the NN decoder at the destination node remains the same as Section 2.2.3.

6.4 Training of Proposed AE Framework with Hyper-Parameter Setting

The training process of the AE-based FD-AF relay network is shown in Fig. 6.4.1, which aims to maximize the chances of reconstruction of the intended signal \mathbf{u}_s by learning the NN's optimization parameters (θ_s, θ_d) .

Please note we train the proposed AE framework using binary CE loss and the training loss optimization remains the same as Section 2.4. Also, the training and testing procedure remains the same as Section 2.5.1 and Section 2.5.2, respectively,

6.4.1 Training and Testing – Dataset Creation and Methodology

In contrast to training dataset creation methods for AE frameworks in [12]–[40] or previous Chapters 2–5, where HD transmission mode is considered with no training dataset creation methodology in the presence of RSI for a FD network. In this chapter, we de-

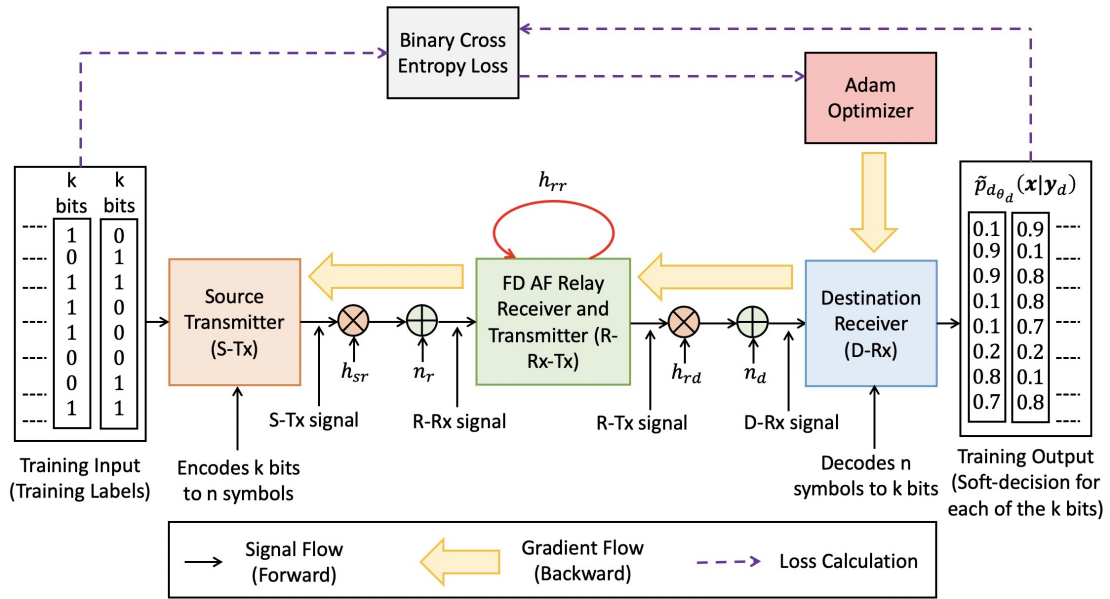


Figure 6.4.1: Block diagram of training for the proposed AE-based end-to-end learning system.

sign a single training dataset such that the AE framework can generalize well for varying testing RSI or SNR values, detailed as follows:

- *For Varying RSI* – For any given rate R or (n, k) , we create a training dataset with S_{Train} samples with fixed transmit SNR $E_b/N_0 = 30$ dB and multiple RSI levels $\sigma_{rr}^2 = \{-60, -20, 0, 20\}$ dB. Then we train a single AE framework until the convergence as detailed in Remark 19 later. Then, we test the trained AE framework for varying RSI levels $\sigma_{rr}^2 = [-60, 20]$ dB.
- *For Varying Transmit SNR* – For any given rate R or (n, k) , we create a training dataset with S_{Train} samples with fixed RSI $\sigma_{rr}^2 = 0$ dB and multiple transmit SNR $E_b/N_0 = \{3, 10, 23, 28, 38\}$ dB. Then we train a single AE framework until the convergence as detailed in Remark 19 later. Then, we test the trained AE framework for varying transmit SNRs $E_b/N_0 = [0, 30]$ dB.

6.4.2 NN Architectures

For generalizability, in this chapter, we propose the same NN architecture for both AE-based BCM and d-BCM designs as shown in Table 6.4.1 with only difference in the Lambda layers $L_L(y_d)$ at the NN decoder. In general, the radio transformer network (RTN) networks are used for d-BCM designs as a means to estimate the channel [12], as also considered in previous Chapters 2–4 performing AE-based d-BCM design in HD transmission mode. But, by experiments, we find that due to the presence of RSI at the FD-AF relay node, the proposed RTN is improving the AE’s performance for BCM design

Table 6.4.1: NN architectures at different nodes in the proposed AE framework.

NN Encoder		NN Decoder		RTN in Lambda layer	
Neurons	Remark	Neurons	Remark	Neurons	Remark
$\delta_1 = k$	Input (\mathbf{u}_s)	$\delta_1 = 2n$	Input (\mathbf{y}_d)	$\delta_1 = 2n$	Input (\mathbf{y}_d)
$\delta_2 = 256$	$\sigma_1 = \text{Tanh}$	$\delta_2 = 2n$	Lambda layer (\mathbf{L}_L)	$\delta_2 = 2n$	Channel h_{sr} equalization
$\delta_3 = 128$	$\sigma_2 = \text{Tanh}$	$\delta_3 = 1024$	$\sigma_1 = \text{Tanh}$	$\delta_3 = 2n$	Channel h_{rd} equalization
$\delta_4 = 64$	$\sigma_2 = \text{Tanh}$	$\delta_4 = 512$	$\sigma_2 = \text{Tanh}$	$\delta_4 = 16$	$\sigma_1 = \text{Tanh}$
$\delta_5 = 2n$	$\sigma_3 = \text{Linear}$	$\delta_4 = 256$	$\sigma_3 = \text{Tanh}$	$\delta_5 = 2n$	$\sigma_2 = \text{Linear}$
$\delta_6 = 2n$	Power normalization (\mathbf{P}_N)	$\delta_5 = 64$	$\sigma_4 = \text{Tanh}$	$\delta_6 = 2n$	Add $2n$ Input (\mathbf{y}_d) and $2n$ output of previous layer
		$\delta_6 = k$	$\sigma_5 = \text{Sigmoid}$		
$\delta_7 = 2n$	Output (\mathbf{x}_s)	$\delta_7 = k$	Soft output ($\tilde{p}_{g\theta_d}(\mathbf{u}_s \mathbf{y}_d)$)		

instead of the d-BCM design. Thus, in contrast to the Chapters 2–4 performing BCM and d-BCM designs for HD transmissions without and with RTN in the NN decoder, respectively. In this chapter, for the FD mode transmission, we employ an RTN in the Lambda layers as proposed in Table 6.4.1 for the AE-based BCM design, and for the AE-based d-BCM design, we consider no RTN or Lambda layers in the NN decoder, i.e., \mathbf{y}_d directly becomes the input to the NN decoder. Please see details in Appendix 6.B.

6.4.3 Hyper-Parameter Settings

The AE framework is implemented in Keras [43] with TensorFlow [44] as backend. For training we utilize SGD with Adam optimizer [42], where the weights are initialized using Glorot initializer [45]. We consider $S_{\text{train}} = 3 \times 10^5$ training samples, $S_{\text{test}} = 10^5$ testing samples and $\tau = 0.001$ learning rate. Also, by parameter searching, we note that smaller batch-size (B) and fewer epochs (E) leads to better performance for BCM design in comparison to d-BCM design. Thus, we keep $B = 128, E = 15$ for performing BCM design and $B = 6000, E = 60$ for performing d-BCM design. This is because while performing the d-BCM design in the absence of CSI knowledge, large batch size provides the AE framework with sufficient samples at low probability region and more epochs helps the AE framework in estimating and removing the channel impairments.

6.5 Convergence of the Proposed AE Frameworks for FD-AF Relay Networks

In Chapter 2 and Chapter 3, we investigated the convergence of the training of the AE frameworks for P2P and relay networks operating in HD transmission mode, respectively. In this section, we build on the same method by analyzing the convergence of proposed AE frameworks operating in an FD transmission mode in the presence of RSI.

6.5.1 Information-Theoretic View of Convergence of Proposed AE Frameworks

We demonstrate the impact of varying SNR and RSI levels and the presence/absence of CSI knowledge on the convergence of the training of the proposed AE framework for the FD-AF relay networks by giving the following Propositions.

Remark 16 *The training of the AE framework, of sufficiently large block length (n), converges to a global minima above a minimum required SNR and below a maximum RSI level.*

Remark 17 *The CSI knowledge is helpful to reach near to the global minima faster, i.e. the convergence of the training of AE while performing BCM design is faster than performing d-BCM design. Please note this remark remains the same as Remark 4 for AE-based BCM and d-BCM designs in a P2P network.*

Proof: Please note the *Proof* of Remark 16 and Remark 17 remains the same as the *Proof* of Propositions 2–4. This is because we are minimizing the same binary CE loss and thus the process of obtaining the estimated MI in (2.19) will also remain the same. However, herein we aim to analyze the impact of the RSI on the training convergence of the AE framework. Thus, we provide a brief Proof below.

For the sake of clarity, we summarize the estimated MI (\mathcal{I}) for the FD-AF relay network below

$$\mathcal{I} := I_{s_{\theta_s}}(\mathbf{u}_s; \mathbf{y}_d) - D_{\text{KL}}\left(p_{s_{\theta_s}}(\mathbf{u}_s|\mathbf{y}_d) \parallel \tilde{p}_{d_{\theta_d}}(\mathbf{u}_s|\mathbf{y}_d)\right) = H(\mathbf{u}_s) - \mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d) \quad (6.8)$$

where $I_{s_{\theta_s}}(\mathbf{u}_s; \mathbf{y}_d)$ is the MI between the input bits \mathbf{u}_s and the received signal at the destination node \mathbf{y}_d with parameters $\boldsymbol{\theta}_s$, $D_{\text{KL}}\left(p_{s_{\theta_s}}(\mathbf{u}_s|\mathbf{y}_d) \parallel \tilde{p}_{d_{\theta_d}}(\mathbf{u}_s|\mathbf{y}_d)\right)$ denotes the Kullback-Leibler (KL)-divergence loss between the true distribution $p_{s_{\theta_s}}(\mathbf{u}_s|\mathbf{y}_d)$ at the NN encoder of the source node and the learnt distribution $\tilde{p}_{d_{\theta_d}}(\mathbf{u}_s|\mathbf{y}_d)$ at the NN decoder of the destination node, with parameters $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_d$, respectively, $H(\mathbf{u}_s)$ is entropy of the input bits \mathbf{u}_s at the source node, and $\mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d)$ is the binary CE loss with NN optimization parameters $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_d$.

Similar to (2.19) (in Chapter 2), since the first term on R.H.S, $H(\mathbf{u}_s)$ in (6.8), remains a constant, thus the changes in the estimated MI in (6.8) only depends on the binary CE loss term $\mathcal{J}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_d)$.

Now, by simulations, we analyze the convergence of the training of AE frameworks performing BCM and d-BCM design in a FD transmission mode. In particular, we train a separate AE performing BCM and d-BCM design for each SNR or RSI level using the NN architecture and hyper-parameter settings as described in Sec. 6.4, once the AE

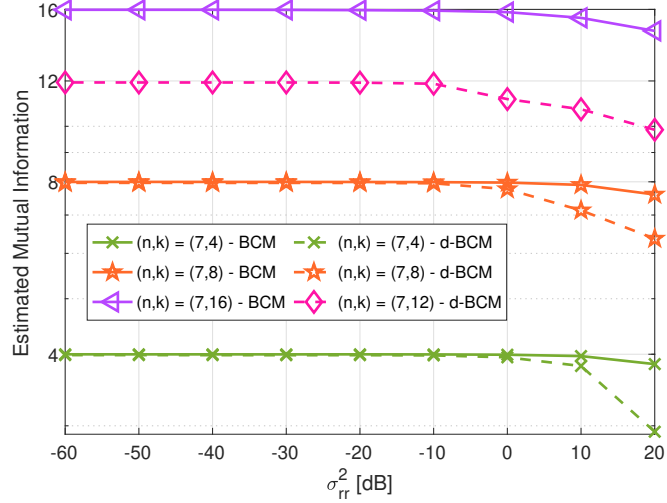
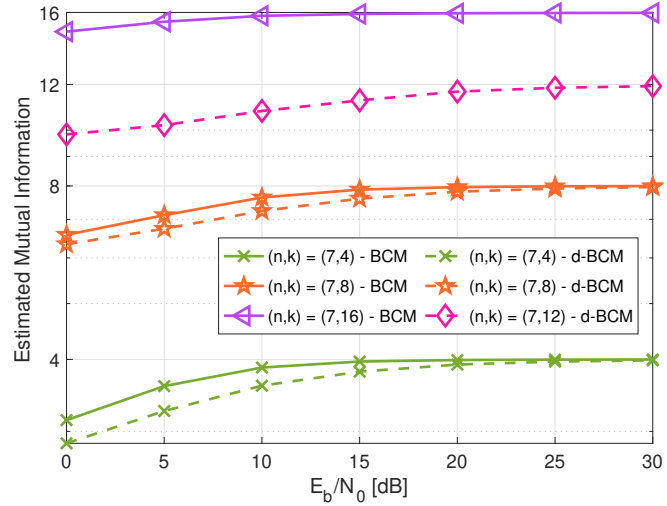

 (a) Varying RSI for fixed $E_b/N_0 = 30$ dB.

 (b) Varying SNR for fixed $\sigma_{rr}^2 = -20$ dB.

Figure 6.5.1: Estimated mutual information for proposed AE-based BCM and d-BCM designs, indicating the convergence of the AE frameworks for varying RSI and SNR levels.

is trained we note the validation CE loss ($\mathcal{J}(\theta_e, \theta_d)$) at the last epoch, and obtain the estimated MI (\mathcal{I}) as described in (6.8). Specifically, we train the proposed AE for fixed SNR $E_b/N_0 = 30$ dB and varying RSI in Fig. 6.5.1a, and for fixed RSI $\sigma_{rr}^2 = -20$ dB and varying SNR (E_b/N_0) in Fig. 6.5.1b. For greater insights, we also vary the rate $R = k/n$ [bits/channel reuse] and keep block size sufficiently large, $n = 7$. We will discuss on block size in detail later in Section 6.6.

In Fig. 6.5.1, we can see that as the RSI decreases or SNR increases, the estimated MI increases until it reaches the upper bound of k . Directly from (6.8), it suggests that KL-divergence loss approaches 0 making $I_{s_{\theta_s}}(\mathbf{u}_s; \mathbf{y}_d) = H(\mathbf{u}_s)$. Thus, the training of AE-based FD-AF relay networks converges to a global minima above a minimum required SNR (in Fig. 6.5.1b) and below a maximum RSI level (in Fig. 6.5.1a). Further, the estimated MI of the AE framework performing d-BCM design converges to the upper

bound at a higher transmit SNR and lower RSI levels. Thus, CSI knowledge is helpful to reach near to the global minima faster (in Fig. 6.5.1). In other words, AE performing BCM design converges faster near to global minima compared to the AE performing d-BCM design. It is important to note that we can't find the global minima of the NN parameters with respect to the binary CE loss. But, surprisingly we don't need to find the global minima. Empirically, the authors in [50, 51] found that despite the non-convexity, the local minima's are rare and they are all very similar to each other and the global minima. Interested readers, please refer to the theoretical insights presented in [50, 51].

■

Remark 18 *As seen in Remark 16 and Remark 17, the proposed AE converges above a minimum SNR and below a maximum RSI. For this, we need to train a different AE framework for each SNR and/or RSI level, which is impractical. For practical purposes, we propose to train a single AE framework on varying SNR or RSI levels (in Sec. 6.4.1), such that the proposed AE can generalize well for the varying SNR and RSI levels. As a result, although the AE's estimated MI never reaches the upper bound because of training on low SNR or high RSI, it enables the AE to generalize well in varying testing scenarios.*

6.5.2 Necessary Conditions for AE Framework's Convergence

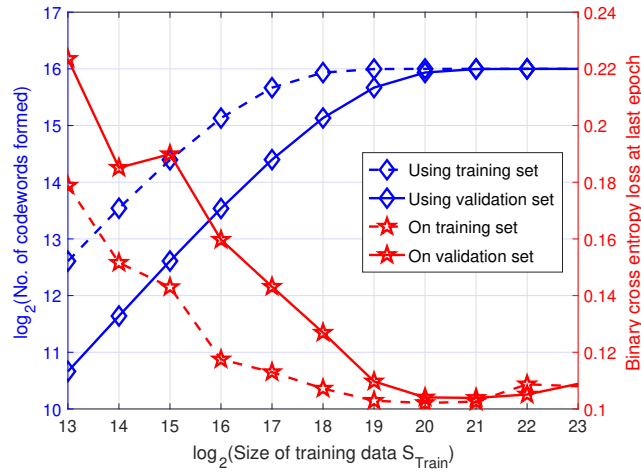
In Section 2.8, we investigated the necessary conditions for the AE framework's convergence in a P2P network operating in an HD mode. In this section, we analyze the same necessary conditions for the AE framework's convergence in an AF relay network operating in the FD mode in the presence of the RSI. For the sake of clarity, we detail the Remark 5 as Remark 19, below

Remark 19 *For any given hyper-parameter settings and rate $R = k/n$, for sufficiently large block length (n), the two necessary conditions for the convergence of training of the AE frameworks performing BCM designs are detailed as follows:*

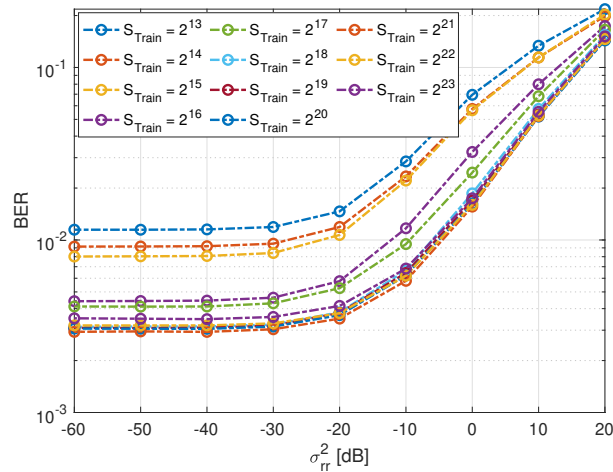
C_1 : *The validation CE loss of the AE framework has converged.*

C_2 : *The NN encoder of the AE designs 2^k codewords.*

Remark 20 *For any given hyper-parameter settings and rate $R = k/n$, with sufficiently large block length (n), the BER performance of the AE framework performing BCM design converge with double training samples because of RSI in a FD transmission mode as compared to the HD transmission mode.*



(a) Codewords formed by the NN Encoder and binary CE loss on training and validation sets.



(b) BER analysis on test set for varying number of Training samples.

Figure 6.5.2: Proof of the C_1 (second part) and C_2 that AE converges once it designs 2^k codewords.

Proof: Please note the *Proof* of Remark 19 and Remark 20 remains the same as the *Proof* of Remark 5, 6. Moreover, we have already shown in Section 6.5.1 that the proposed AE converges above a minimum required SNR and below a maximum RSI level. Thus, we focus on showing the relationship between the convergence of the AE frameworks with the training samples below.

Please note that the condition C_1 in Remark 12 comprises of two parts, detailed in Remark 5. We now empirically prove the conditions C_1 (second part) and C_2 below. For example, we train an AE performing BCM design for $R = 16/7$ in Fig. 6.5.2 for varying training data size $S_{\text{Train}} = \{2^{13}, \dots, 2^{23}\}$ of fixed SNR $E_b/N_0 = 30$ dB. Specifically, we divide the S_{Train} training samples into 4 : 1 ratio of training set S_T and validation set S_V . Then, we train the AE on S_T and determine the number of codewords formed by the NN encoder and the binary CE loss at the last epoch (15th epoch) on S_T and S_V . Lastly, we

determine the BER using the testing samples S_{Test} .

In Fig. 6.5.2a, we can see that as the training dataset increases, the number of codewords formed by the NN encoder of the trained AE on the training and validation sets increases until it becomes 2^{16} codewords, each representing one of the possible 2^{16} combinations. Moreover, the NN encoder forms these 2^{16} codewords on the 2^{18} and 2^{21} training samples using the training and validation sets, respectively. Furthermore, in Fig. 6.5.2a, we can see that the binary CE loss, noted at the last (15th) epoch of training, reduces as the training dataset increases and converges for training and validation sets at 2^{21} training samples.

In Fig. 6.5.2b, we can see that as the training dataset increases, the performance of the proposed AE on the unseen testing samples improves. In contrast, when the training dataset size starts becoming greater than 2^{18} , then the performance improvement of the proposed AE starts converging because 2^{16} codewords are created by the NN encoder of the AE on the training set S_T .

Thus, for any given hyper-parameter settings, we at least need 2^{k+2} training samples to ensure the AE creates 2^k codewords, the validation CE loss has converged, and the AE's performance converges to its maximum potential of decoding the 2^k possible classes, and no more training samples can help in improving the BER performance. ■

Please note that in Remark 6 for P2P network in HD transmission mode, we found that the AE framework performing BCM design converge with at least 2^{k+1} training samples. However, in the FD transmission mode, the BER performance of the AE framework performing BCM design converges with at least 2^{k+2} training samples. This shows that RSI leads to doubling the number of training samples required for the AE's BER performance convergence.

6.6 Analyzing the Observations of AE-based Block Coded Modulation

To understand the generalizability of the observations detailed in Section 2.9, we analyzed the distinct observations of AE-based BCM and d-BCM for P2P networks operating in HD mode. Employing the same methodology, we analyze the AE-based BCM for AF relay networks operating in FD mode. Please note for brevity, we only consider BCM by AE frameworks because, as shown in Section 2.9, both the BCM and d-BCM exhibit similar trends. Throughout this section, we train the proposed AE frameworks for various rates $R = k/n$ or combinations of (n, k) , where $n \in \mathcal{N} = \{1, 3, 5, 7, 10\}$ and $k \in \mathcal{K} = \{1, 4, 8, 12, 16\}$ using the training settings in Section 6.4 and until convergence using

Remark 19. Once trained, the NN encoder becomes deterministic. Thus, if we input any k bits to the NN encoder of the trained AE, we obtain the same n complex baseband symbols as output every time, representing a codeword for the k input bits. Now, we can obtain all the possible codeword from the NN encoder using all the possible combinations of k input bits.

For example, we analyze the applicability of Observations – 1, 2, 7, 8, 9, 12 detailed for the BCM design in P2P network as Remarks below. In particular, the BCM design by the proposed AE-based FD-AF relay networks exhibit the following five distinct properties, noted as remarks below, for any (n, k) , as:

Remark 21 *AE framework designs 2^k codewords in $2n$ -dimensional space.*

This property remains straightforward and similar to Observation – 1 of the P2P networks.

Remark 22 *1. As the block, length increases the minimum Euclidean distance between any of the possible codewords increases.*

2. When the number of codewords becomes extremely large, the minimum Euclidean distance between any two codewords follows a Gaussian distribution for a sufficiently large block length (n).

3. As the block length increases, the Euclidean distance between the codewords concentrate to the average Euclidean distance².

We evaluate the minimum Euclidean distance³-based *Observations – 2, 7, 8* for the HD P2P networks proposed in Sec. 2.9 as the aforementioned Remark 22 for the FD AF relay networks in presence of the RSI. We follow the similar process as HD P2P networks to obtain – (i) the minimum Euclidean distance between all the 2^k designed codewords ($d_{E_{\min}}$) in Fig. 6.6.1, (ii) the impact of increasing minimum Euclidean distance $d_{E_{\min}}$ on the BER performance of the proposed AE framework in Fig. 6.6.2, and (iii) plot a histogram of the minimum Euclidean distance between each a^{th} codeword and its closest v^{th} codeword for varying $(n \in \mathcal{N}, k \in \mathcal{K})$ in Fig. 6.6.3. In Fig. 6.6.1–6.6.3 for FD-AF relay networks, we see similar trends for the designed BCM as HD P2P networks. Thus, proving that the minimum Euclidean distance-based observations proposed for P2P networks also hold for complicated scenarios, such as FD relaying networks.

Remark 23 *The codes designed by the AE framework are spherical codes.*

²Please note the average Euclidean distance is the mean of the minimum Euclidean distance between any codeword and its closest codeword.

³As detailed in Definition 16.

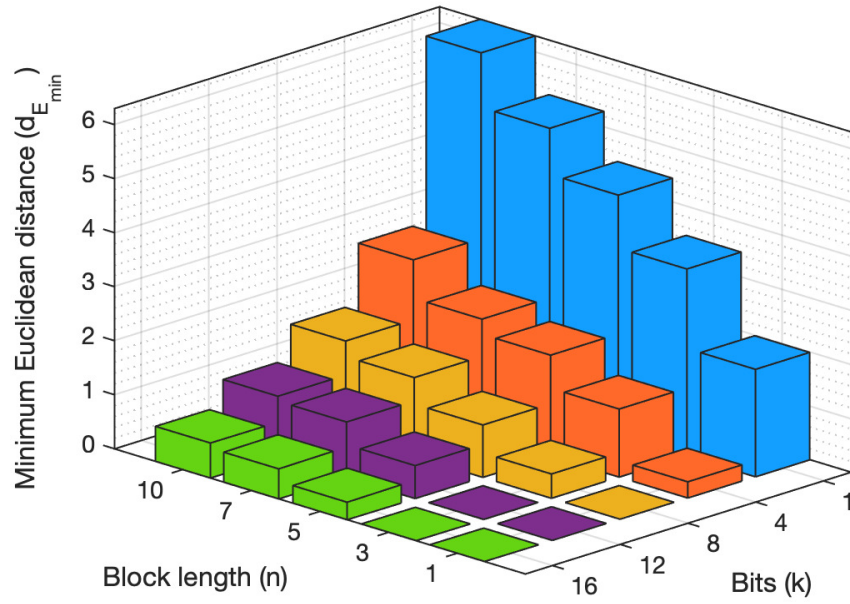


Figure 6.6.1: Minimum Euclidean distance $d_{E_{\min}}$ for varying (n, k) .

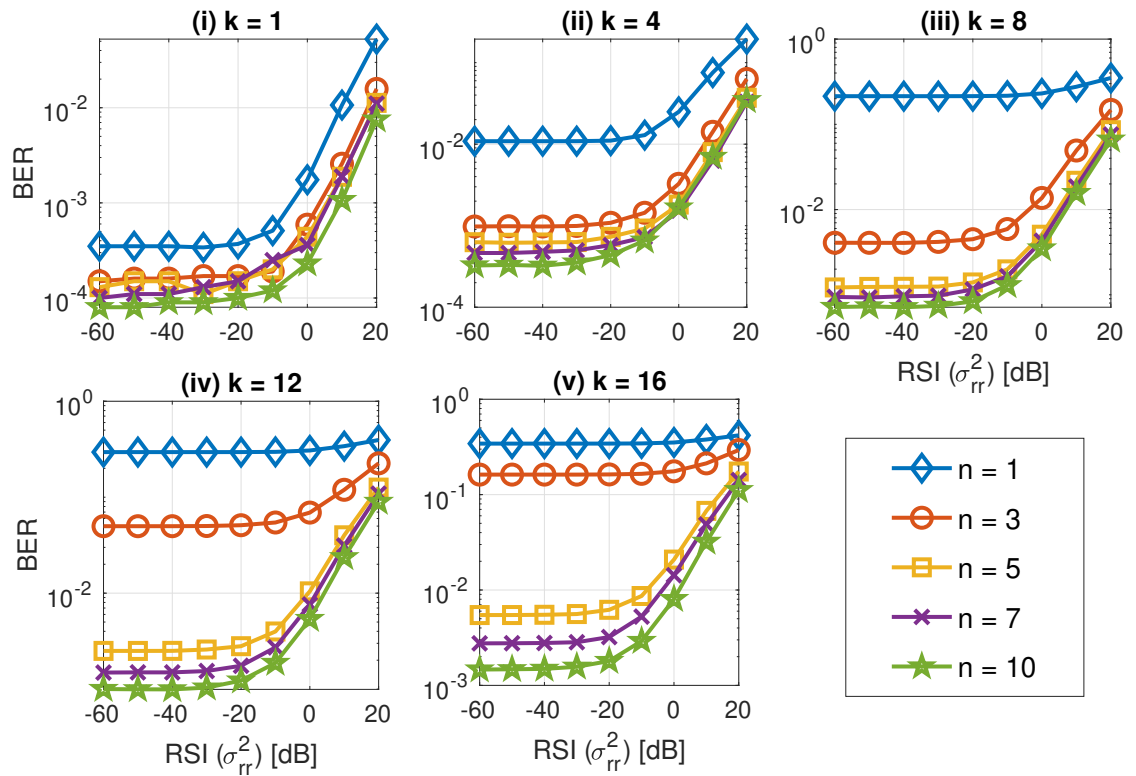


Figure 6.6.2: BER versus RSI for varying (n, k) .

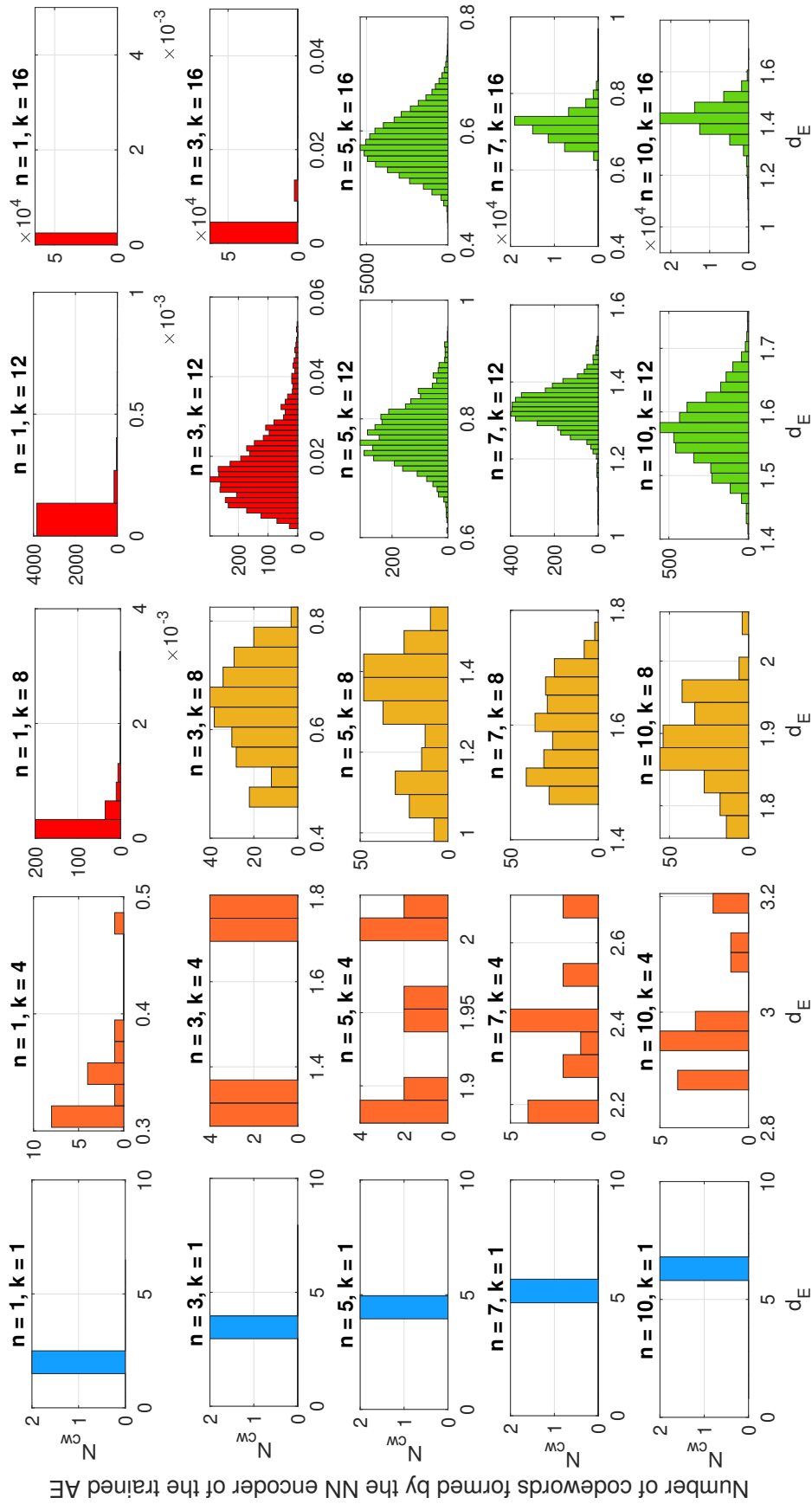


Figure 6.6.3: Minimum Euclidean distance d_E between each a^{th} codeword and its closest v^{th} codeword.

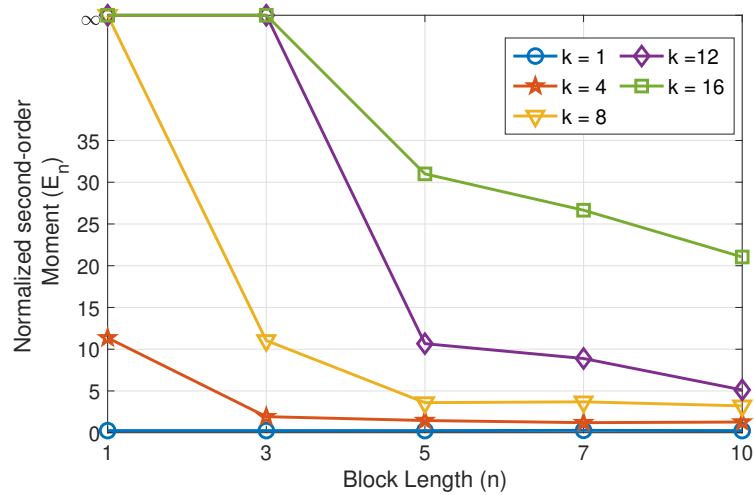


Figure 6.6.4: Packing density.

By simulations, we find that the proposed AE creates BCM design with fourth-order moment or Kurtosis (as detailed in Definition 18) being equal to 1, i.e., ‘Spherical codes’ are being created as $\chi = 1$, i.e. all the 2^k codewords for all the varying ($n \in \mathcal{N}, k \in \mathcal{K}$) scenarios have equal norm, which is again similar to the *Observation - 12* for the HD P2P networks.

Remark 24 *The packing density improves as the rate R decreases.*

In Fig. 6.6.4 we analyze the packing density E_n (as detailed in Definition 17) with varying rate $R = k/n$. We can see that the packing density improves for the AE-based BCM as the block-length (n) increases or the input bits (k) decreases, for all ($n \in \mathcal{N}, k \in \mathcal{K}$), which is again similar to the *Observation - 9* for the HD P2P networks, showing the merits of observations proposed in Chapter 2.

Observation – 13: As the block-length increases the average Hamming distance between codewords increases.

Proof: Determining the Hamming distance between the two closest codewords is a simple task if the symbols representing the codewords in a constellation diagram form a grid like structure with equal Euclidean distance, such as conventional QPSK modulation. We can simply utilize the Euclidean distance to determine the closest points and determine the Hamming distance for any symbol. However, the AE framework does not design codewords in an exact grid like structure with equal Euclidean distance between the codewords. Specifically, using the *Observation - 2*, we know that the 2^k codewords are being designed in the $2n$ -dimensional space that have the minimum Euclidean distance between the a^{th} codeword and its closest v^{th} codeword as Gaussian distributed, where $\{a, v\} = \{1, \dots, 2^k\}$ and $v \neq a$. Thus, the distance between any two codewords

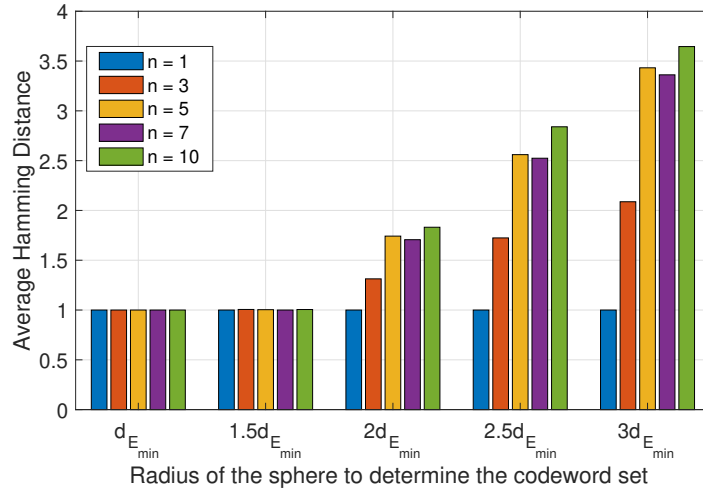


Figure 6.6.5: Average Hamming distance.

is different and does not follow a grid-like structure, hence we cannot directly utilize the minimum Euclidean distance to determine the Hamming distance between two closest codewords, and we utilize the average Hamming distance as the metric. Hence in this chapter, we firstly determine the minimum Euclidean distance $d_{E_{\min}}$ of the 2^k codewords using (2.20). Then, for each a^{th} codeword, we determine all the codewords within the sphere with radius given by the minimum Euclidean distance $d^a \leq d_{E_{\min}} + \xi$, such that $\xi \geq 0$ and represent these codewords by a set \mathcal{S}^f . We then determine the average Hamming distance⁴ for each a^{th} codeword and all the codewords in its corresponding set \mathcal{S}^a [100, 101], as

$$d_H^{\text{avg},a} = \sum_{v \in \mathcal{S}^a} \frac{d_H(a, v)}{|\mathcal{S}^a|} \quad (6.9)$$

where $d_H(a, v)$ denotes the Hamming distance between codeword a and v , and $|\mathcal{S}^a|$ is the cardinality of the set \mathcal{S}^a . Now, we can determine the average Hamming distance for all the $a \in \{1, \dots, 2^k\}$ codewords as

$$d_H^{\text{avg}} = \frac{1}{|d_H^{\text{avg},a} > 0|} \sum_{a=1}^{2^k} d_H^{\text{avg},a} \quad (6.10)$$

where $|d_H^{\text{avg},a} > 0|$ is the number of non-zero elements in $\mathbf{d}_H^{\text{avg}}$. For fixed input bits $k = 8$, we determine the average Hamming distance in (6.10) for varying block lengths $n \in \mathcal{N}$ and varying $\xi = \{0, 0.5, 1, 1.5, 2\}$ in Fig. 6.6.5. As expected, as the radius of the sphere ($d_{E_{\min}} + \xi$) to determine the codeword set \mathcal{S}^a increases the average Hamming distance d_H^{avg} increases. Interestingly, as the block-length (n) increases the average Hamming distance

⁴Please note the average Hamming distance is the mean of the Hamming distance between any codeword and its closest codeword in the set \mathcal{S}^f .

d_H^{avg} also increases, because in Fig. 6.6.1 we see that as the block-length (n) increases the minimum Euclidean distance between the codewords $d_{E_{\min}}$ is also increasing. ■

6.7 Numerical Results

In this section, we evaluate the performance of the proposed AE-based BCM and d-BCM designs for FD-AF relay networks and conventional HD-AF and FD-AF relay networks. As this is the first time NN-based AE framework is proposed in the context of FD networks, for a fair comparison, we consider the conventional FD-AF relay networks as a benchmark, wherein we utilize traditional modulation techniques and (7, 4) Hamming code as a baseline error correction code, with the MLD decoding detailed in (6.5). Also, we utilize RBF channels such that it remains constant for $n = 7$ transmissions only. For the sake of clarity, we show the performance gains of employing – (i) conventional FD-AF relay node over a NN-based relay node in Appendix 6.A and (ii) inclusion of an RTN in BCM design instead of d-BCM design in Appendix 6.B.

6.7.1 AE-based d-BCM Design

In Fig. 6.7.1, we analyze the BER of the proposed AE-based d-BCM without the CSI knowledge. In Fig. 6.7.1a, 6.7.1c, 6.7.1e, we fixed transmit SNR $E_b/N_0 = 30$ dB and vary the RSI, for varying input bits (k)⁵. We can see that for small RSI ($\sigma_{rr}^2 \leq -30$ dB) the BER performance of – (i) conventional FD-AF and HD-AF relay networks’ becomes same and (ii) proposed AE-based d-BCM converges, because the RSI becomes negligible to impact the signal at the FD-AF relay node. Furthermore, the proposed AE-based d-BCM design outperforms the conventional FD-AF relay networks for all varying rates (R) and RSI levels. In Fig. 6.7.1b, 6.7.1d, 6.7.1f, we fixed the RSI at $\sigma_{rr}^2 = 0$ dB and vary the transmit SNR, for varying (k)³. The conventional FD-AF relay networks is not able to decode the signals even for $E_b/N_0 = 30$ dB as the RSI is high ($\sigma_{rr}^2 = 0$ dB), but the proposed AE-based d-BCM is able to decode the signals and the BER reduces with increasing SNR.

We explain the reasons for the gains achieved by the AE-based d-BCM as follows – the proposed AE can design 2^k codewords in $2n$ -dimensional space with automatic bit-labeling by maximizing the bit-wise MI. Furthermore, the AE aims to learn these d-BCM designs to remove the deteriorating impacts of RSI, RBF channels, and AWGN at the nodes by the proposed end-to-end training until convergence using Remark 19. This leads to the maximization of the minimum Euclidean distance and minimum average Hamming

⁵For fixed $n = 7$, we keep k as 4, 8, and 12, that corresponds to d-BPSK, d-QPSK and d-8PSK modulations designs in conventional networks with (7, 4) Hamming coding.

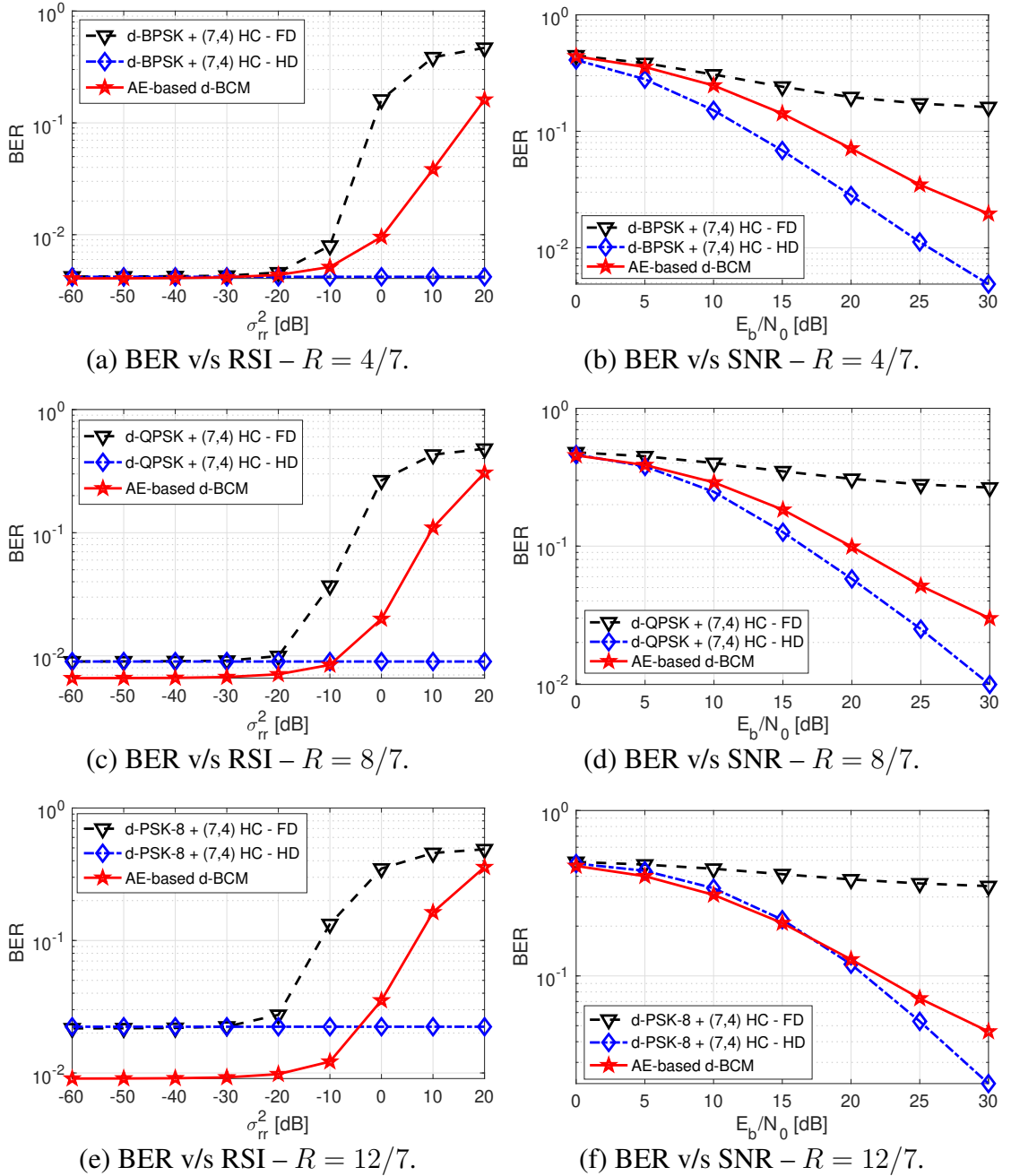


Figure 6.7.1: Performance evaluation for AE-based d-BCM for FD-AF relay networks.

distance as detailed in Sec. 6.6 for the designed codewords and thus achieve improvement in the BER performance.

In Fig. 6.7.1a, 6.7.1c, 6.7.1e, the proposed AE is able to design the d-BCM for 2^k codewords in $2n$ -dimensional space with the observations detailed in Sec. 6.6, leading to the BER performance for proposed AE almost similar for any rate $R \leq 12/7$. Thus, as the modulation order or rate increases the proposed AE can even outperform the conventional HD-AF relay networks even for higher RSI, i.e. at -10 dB (for $k = 8$) and -5 dB (for $k = 12$) for $n = 7$. Due to similar reasons, in Fig. 6.7.1b, 6.7.1d, 6.7.1f, the AE's BER performance becomes closer to the conventional HD-AF relay networks as the modulation order or rate increases, indicating that the proposed AE almost removes the impact of the RSI even in the absence of CSI and very high RSI levels.

6.7.2 AE-based BCM Design – Perfect CSI Knowledge

In Fig. 6.7.2, we analyze the BER performance of the proposed AE-based BCM with perfect CSI knowledge. For varying $(k)^6$, in Fig. 6.7.2a, 6.7.2c, 6.7.2e we fix transmit SNR $E_b/N_0 = 30$ dB and vary the RSI, and in Fig. 6.7.2b, 6.7.2d, 6.7.2f we fix $\sigma_{rr}^2 = 0$ dB and vary the SNR. We see similar BER performance trends for BCM as d-BCM in Sec. 6.7.1. Except for low RSI $\sigma_{rr}^2 \leq -30$ dB, the AE-based d-BCM design outperformed the HD-AF relay networks. Also, unlike AE-based d-BCM design, the proposed AE-based BCM design's BER performance tends to deteriorate with increasing modulation order or rate due to the following reasons – in the presence of perfect CSI, the BER performance of the conventional FD-AF and HD-AF relay networks is already very good. Although for the AE-based frameworks, many codewords are being packed together as a spherical code, the advantage the AE-based d-BCM had in tackling the RBF channels effectively than conventional differential schemes is not present here because of the availability of the perfect CSI knowledge. Due to similar reasons, we can see in Fig. 6.7.2b, 6.7.2d, 6.7.2f that BER performance of the proposed AE-based BCM design tends to deteriorate with increasing modulation order or rate.

6.7.3 AE-based BCM Design - Imperfect CSI Knowledge

Now, we analyze the AE's performance in the presence of the channel estimation error. We utilize the linear minimum mean squared error (LMMSE) based channel estimation [102] denoted by $h_{(\cdot)}^\omega \in \mathcal{CN}(0, \sigma_{h^\omega}^2)$ where the error in channel estimation is $e_{(\cdot)} \in \mathcal{CN}(0, \sigma_e^2)$ for both the hops $(\cdot) = \{sr, rd\}$. From the orthogonality principle of the LMMSE we know that the errors in the channel estimation remains mutually indepen-

⁶For fixed $n = 7$, we keep k as 4, 8, and 16, that corresponds to BPSK, QPSK and QAM-16 modulations designs in conventional networks with $(7, 4)$ Hamming coding.

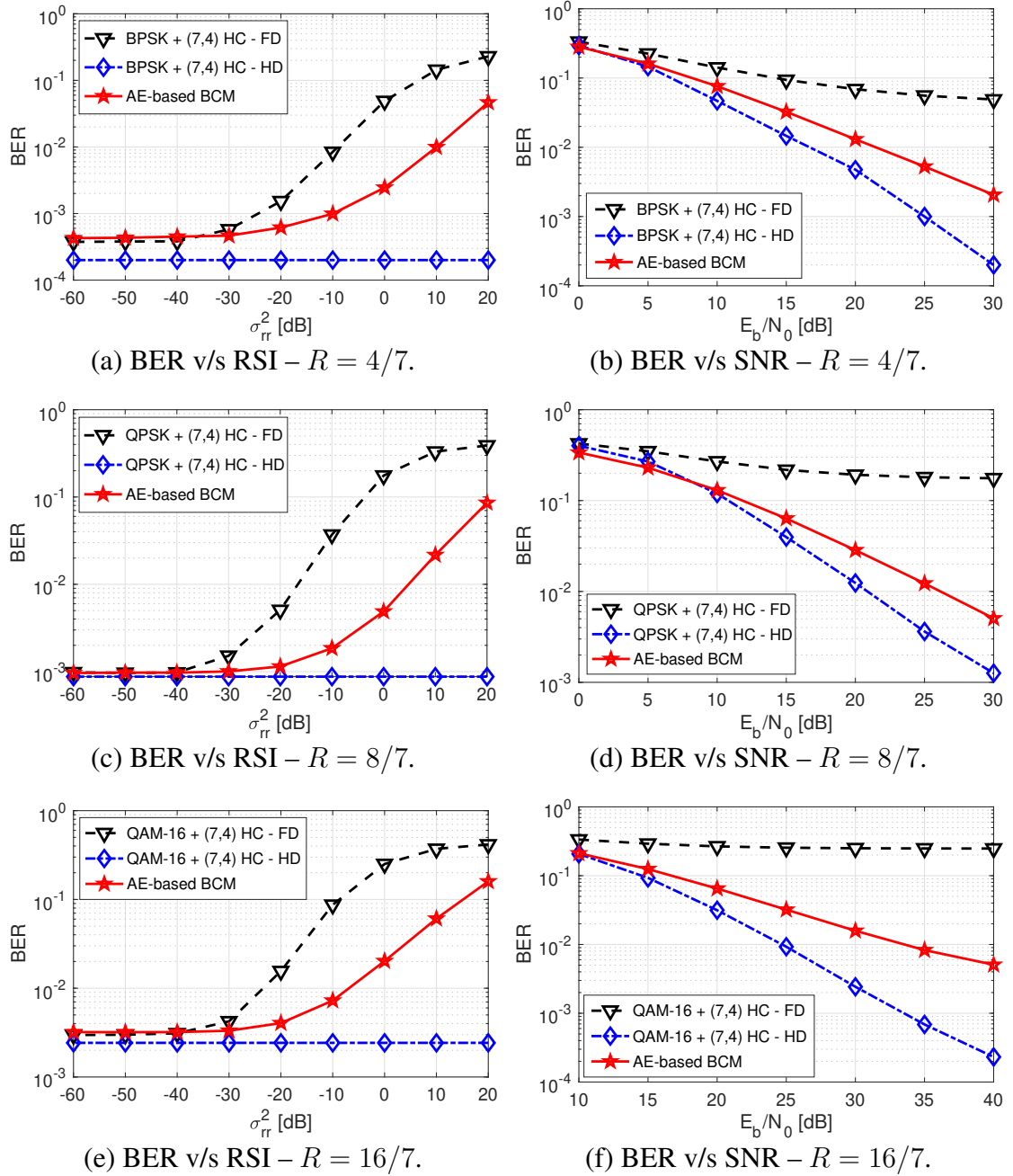


Figure 6.7.2: Performance evaluation for AE-based BCM for FD-AF relay networks.

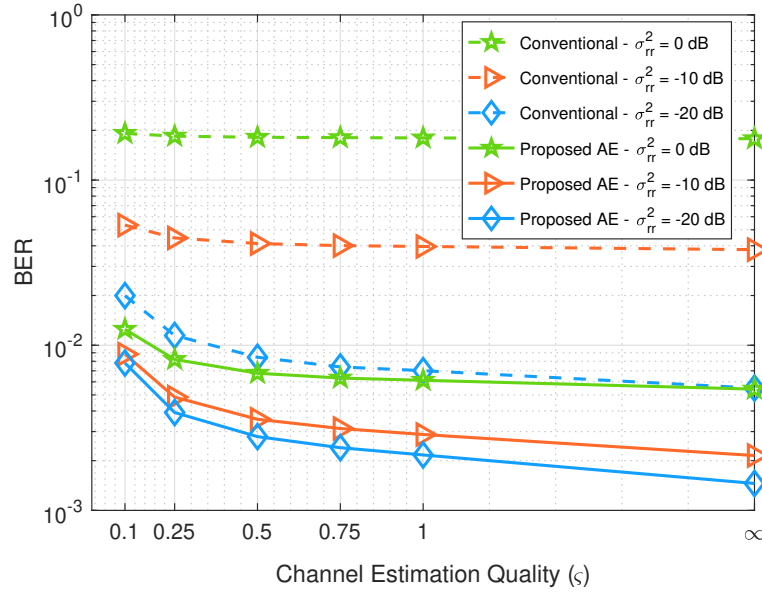


Figure 6.7.3: Impact of the CEQ (ς) on FD-AF relay networks for rate $R = 8/7$.

dent of the estimated channel, thus, we have

$$h_{(\cdot)}^\omega = h_{(\cdot)} + e_{(\cdot)}, \quad \forall (\cdot) = \{sr, rd\} \quad (6.11)$$

We denote the channel estimation quality (CEQ) by ς and assume that the error variance depends on the SNR denoted by γ , such that $\sigma_e^2 = \frac{\sigma_h^2}{1+\varsigma\gamma\sigma_h^2} \approx \frac{1}{1+\varsigma\gamma}$ and $\sigma_{h^\omega}^2 = \frac{\varsigma\gamma\sigma_h^2}{1+\varsigma\gamma\sigma_h^2} \approx \frac{\varsigma\gamma}{1+\varsigma\gamma}$ [102].

In Fig. 6.7.3 we analyze the impact of CEQ (ς) on the proposed AE-based BCM design and conventional QPSK + (7, 4) Hamming code for rate $R = 8/7$ and transmit SNR $E_b/N_0 = 30$ dB. Please note $\varsigma = 0$ indicates completely erroneous channel whereas $\varsigma = \infty$ denotes perfect channel estimation. To create an AE that remains unaffected of the varying channel estimation errors, we train a single AE framework consisting of S_{train} samples from varying $\varsigma = \{0.1, 0.5, 1, \infty\}$ until the convergence as detailed in Remark 19 and test on unseen S_{test} samples of varying CEQs (ς). The proposed AE-based BCM designs outperform the conventional FD-AF relay networks for all the CEQs due to similar reasons as Sec. 6.7.1. The BER performance of the proposed AE framework with almost fully erroneous channel estimation $\varsigma = 0.1$ is better than conventional FD-AF relay networks with perfect channel estimation $\varsigma = \infty$. As the RSI increases, the BER performance improvement by the proposed AE increases. This is because the proposed AE-based BCM is designing 2^k codewords in $2n$ -dimensional space with observations in Sec. 6.6 such that it can handle the impacts of RSI and channel estimation errors effectively.

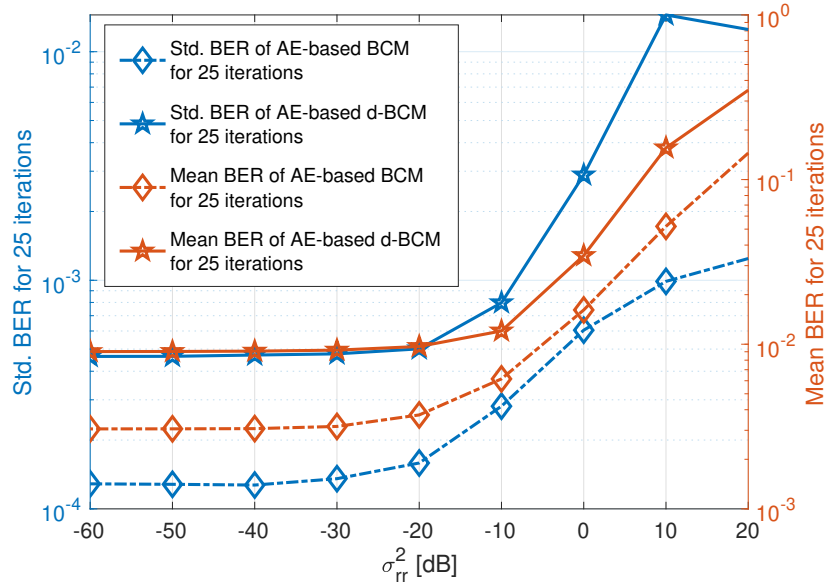


Figure 6.7.4: Reproducibility of AE-based BCM results for $R = 16/7$ and AE-based d-BCM results for $R = 12/7$.

6.7.4 Reproducibility of Proposed AE and d-AE Frameworks

Definition 21 (Reproducibility of AE) An AE is defined to be reproducible, for a given hyper-parameter setting \mathcal{P}^S , if and only if we can reproduce any trained AE model $\mathcal{M}(\theta)$ with a very high probability, such that it does not lead to large variations in BER for different training weight initialization and training-validation samples of the AE.

We analyze the reproducibility by varying training-validation data and weight initialization for training each of the AE performing BCM and d-BCM designs 25 times and reporting the standard deviation and mean of BER in testing data in Fig. 6.7.4. In particular, we evaluate the reproducibility of the BER results obtained for the proposed AE-based BCM and d-BCM for different RSI levels, while we fix rate $R = 16/7$ in AE-based BCM design and fix rate $R = 12/7$ in AE-based d-BCM design, with transmitting SNR $E_b/N_0 = 30$ dB. We can see that the proposed AE frameworks are *highly reproducible* because their standard deviation of 25 BER obtained from 25 different runs lies in the range $10^{-2} - 10^{-4}$. This is because we train the AE until the convergence using Remark 19. Also, as the RSI increases, the variations in BER increases by a factor of two, showing that higher RSI levels negatively impact the reproducibility of the trained AE frameworks in an FD-AF relay network.

6.8 Conclusion

In this chapter, we propose end-to-end learning-based FD-AF relay networks in the presence of the RSI using the AE framework for high transmission rates $R = k/n$. We propose (n, k) AE-based BCM and d-BCM designs depending upon the availability of the CSI knowledge. Further counter-intuitively, in the presence of the RSI in the FD-AF relay networks, we propose to utilize a radio transformer network for the BCM design to improve the NN-based decoding and BER performance. We design a single fixed AE framework that can generalize well on varying testing SNR or RSI levels, outperforming the conventional FD-AF relay networks with remarkable gains and the half-duplex AF relay networks in the absence of CSI knowledge. We analyze the AE's performance in the presence of channel estimation error. We note that the proposed AE framework with almost entirely erroneous channel estimation for moderate RSI still outperforms the conventional FD-AF relay networks with perfect channel estimation. Moreover, we show that the proposed AE framework is highly reproducible for varying training weight initialization and training sample sets as the BER for 25 different training varies by a standard deviation of $10^{-2} - 10^{-4}$ depending on RSI levels in the FD-AF relay node.

With a focus on interpretability, similar to Chapter 2 on HD P2P networks, we determine the estimated mutual information of the proposed AE frameworks for varying SNR and RSI levels. We show that the AE converges above a minimum required SNR and below a maximum RSI, depending on the transmission rate and CSI availability. Furthermore, we provide the necessary conditions for AE's convergence by showing that once the binary cross-entropy validation loss has converged, the NN encoder of AE designs 2^k codewords during the training phase, the AE has converged to its maximum potential of decoding the signal. Lastly, by analyzing the AE-based BCM designs, we determine distinct observations of the designed codewords in $2n$ -dimensional space – (i) AE forms 2^k codewords in $2n$ -dimensional space, (ii) as the block length increases the minimum Euclidean distance between any of the possible codewords increases, and for sufficiently large block length (n) when the number of codewords becomes extremely large, the minimum Euclidean distance between any two codewords follows a Gaussian distribution and the Euclidean distance between the codewords concentrate to the average Euclidean distance, (iii) the packing density improves as the rate R decreases, (iv) the codewords designed by the AE framework are spherical codes, and (v) as the block-length increases the average Hamming distance between codewords increases. Thereby, also showing that the observations of AE-based BCM and d-BCM designs for simple P2P HD transmission mode proposed in Chapter 2 also holds true for complicated dual hop AF relay networks, operating in FD mode. We also introduced a new property relating to average Hamming distance.

Appendix

6.A Conventional FD-AF Relay versus NN-based FD-AF Relay

In this chapter, we propose to utilize conventional FD-AF relay node in the AE framework. In this Appendix, we compare the BER performance of proposed AE framework designed using the conventional FD-AF relay node and AE framework designed using an NN-based FD-AF relay node. In AE works for HD-AF relay networks [38], [57] a NN-based HD-AF relay node is employed, thus we compare three different NN-based FD-AF relay nodes with input as x_r , by varying the layers and activation functions, detailed as follows:

- *NN Relay-1*: 2 (Input: x_r) \rightarrow 2 (*Tanh*) \rightarrow 2 (*Linear*) \rightarrow 2 (P_N)
- *NN Relay-2*: 2 (Input: x_r) \rightarrow $2n$ (*Tanh*) \rightarrow 2 (*Linear*) \rightarrow 2 (P_N)
- *NN Relay-3*: 2 (Input: x_r) \rightarrow $2n$ (*Tanh*) \rightarrow $2n$ (*Tanh*) \rightarrow 2 (*Linear*) \rightarrow 2 (P_N)

where $X(Y)$ indicates there are X neurons in the dense layer and Y denotes activation function. Also, the P_N denotes the power normalization layer, defined as Definition 7, which ensures the symbol power remains 1. In Fig. 6.A.1, we analyze the BER performance of proposed conventional FD-AF relay-based AE and NN-based FD-AF relay-based AE frameworks for $(n, k) = (7, 8)$, $E_b/N_0 = 30$ dB and varying RSI. Clearly, the proposed AE (with conventional FD-AF relay node) outperforms the NN-based FD-AF relay node-based AEs, while having no optimization parameters because there is no NN, thereby also reducing the implementation complexity.

6.B Impact of Including RTN in AE Frameworks

In AE works for the HD-AF relay network (please see *Chapter 3*), an RTN is included in the d-BCM design and excluded in the BCM design. In Fig. 6.B.1, we analyze the impact of including an RTN in the NN decoder of the proposed AE frameworks for $(n, k) = (7, 4)$, $E_b/N_0 = 30$ dB and varying RSI. We can see that including an RTN

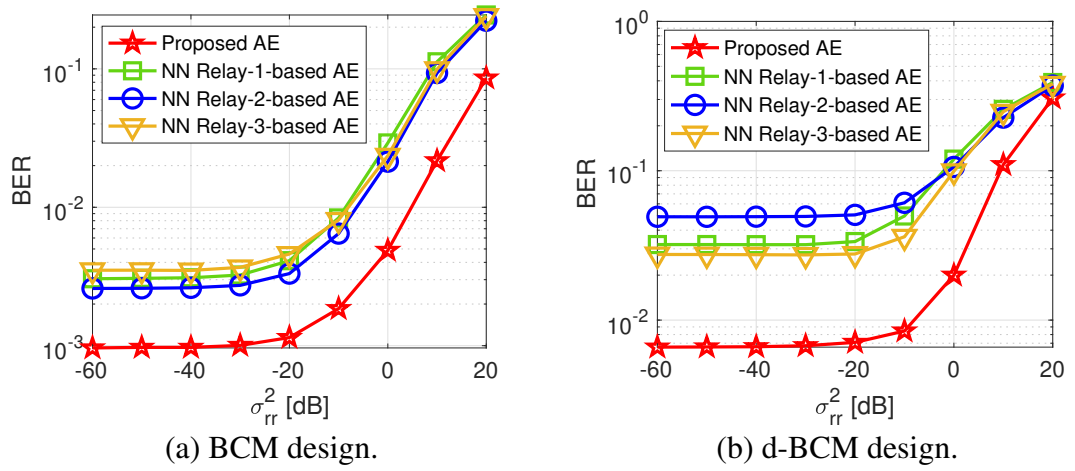


Figure 6.A.1: Comparing proposed conventional v/s NN-based FD-AF relay-based AE frameworks.

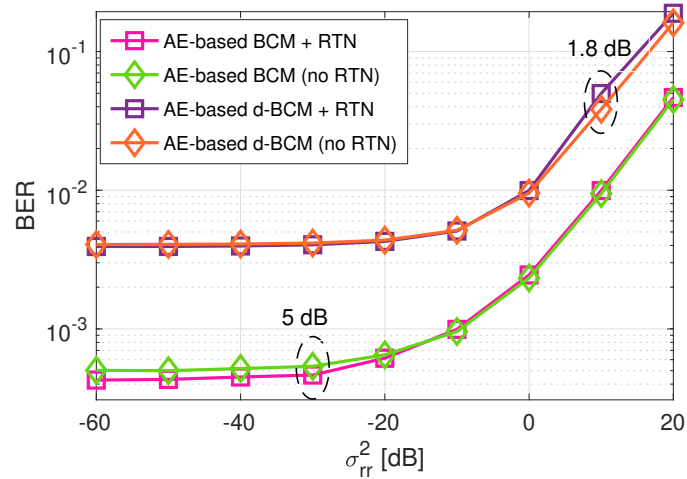


Figure 6.B.1: Impact of including an RTN in the AE-based BCM design and not including RTN in d-BCM design.

in AE performing BCM design helps to improve the BER performance by at least 5 dB for lower RSI ($\sigma_{rr}^2 \leq -20$ dB), whereas including an RTN in AE performing d-BCM design worsens the BER performance by at least 1.8 dB for higher RSI ($\sigma_{rr}^2 \geq 0$ dB). Thus in this chapter, in contrast to Chapters 2–4, we have included an RTN in our proposed AE framework while performing BCM design and do not include an RTN in our proposed AE framework while performing d-BCM design.

Chapter 7

AE-based Decode-and-Forward Relay Networks with Hardware Impairments

7.1 Introduction

Although the amplify-and-forward (AF) relay provides low complexity re-transmission of the amplified signal, it also amplifies the noise at the relay node [25], [24]. Thus, in previous chapters 3–6, we focused on autoencoder (AE)-based block coded modulation (BCM) and differential BCM (d-BCM) designs (for rate $R = k/n$ [bits/channel-reuse]) to minimize the deteriorating impacts of the noise amplification. Traditionally, to remove the noise amplification, decode-and-forward (DF) relay is adopted [22], [56]. In the DF relay protocol, the relay node decodes the received signal, re-encodes the decoded signal, and transmits the encoded signal to the destination node [79, 56, 80, 81, 22, 23]. Moreover, the DF protocol outperforms the AF protocol when the source to relay channel is good enough and has added advantage of using different coding designs at the source and relay nodes. For the sake of clarity, we summarize the one-way DF (OW-DF) and two-way DF (TW-DF) relay networks below

- *One-way decode-and-forward (OW-DF) relay networks* [79, 56, 80, 81] – As shown in Fig. 7.1.1a, in the first phase, terminal node A sends the signal to the relay node. In the second phase, the relay node decodes the received signal and re-encodes the decoded signal (digital-to-digital domain) before re-transmission to terminal node B, which decodes the intended signal.
- *Two-way decode-and-forward (TW-DF) relay networks* [22, 23] – As shown in Fig. 7.1.1b, in the first phase, terminal node A sends its signal to the relay node, which decodes the terminal node A's intended signal. Similarly, terminal node B sends its signal in the second phase, which is decoded by the relay node. Finally, in the third phase, the relay node combines the decoded signals and broadcasts the re-

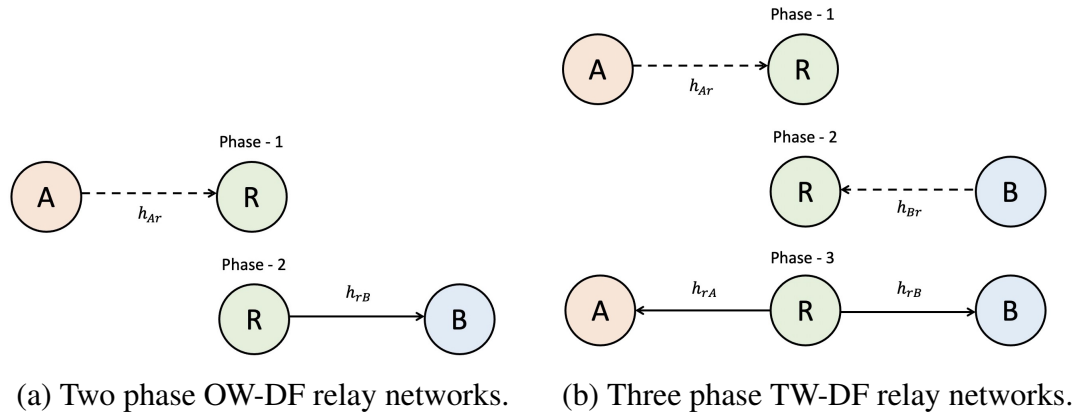


Figure 7.1.1: An illustration of OW-DF and TW-DF relay networks implementing the signal transmission–reception between terminal nodes A and B in two and three phase transmissions.

encoded combined signal to the terminal nodes, each removing its signal, decoding the intended signal from the other terminal node. For the third phase transmission–reception, the two widely employed techniques are detailed as

- *Denoise and superposition forward (DSF) technique* [103] – Herein, the relay node superposes the decoded signals using a factor $\alpha \in (0, 1)$, while the terminal nodes subtract its signal with the knowledge of α to decode the signal of the other terminal node.
- *Decode and XOR forward (DXF) technique* [104] – Herein, the relay node performs bit-wise XOR operation on the decoded signals before re-encoding the signal, while the terminal node performs bit-wise XOR on the decoded signal and its signal to obtain the signal of the other terminal node.

Recently, the AE-based end-to-end learning frameworks have been designed for the OW-DF relaying networks, showing promising bit-error-rate (BER) performance gains [37, 39, 36], our recent work [67], wherein the terminal node A transmits its signal to the terminal node B employing an OW-DF relay node. The first phase has a different AE framework, wherein an NN-based encoder at terminal node A encodes the signal and transmits it to the OW-DF relay node, which decodes the signal using an NN-based decoder. Also, the second phase has a different AE framework, wherein the NN-based encoder at the relay node encodes the signal and transmits it to the terminal node B, which decodes the signal using an NN-based decoder. The OW-DF relay network is investigated using a symbol-wise AE framework [37, 36], [67] and a bit-wise AE framework for a cooperative non-orthogonal multiple access, wherein terminal node A operates as a base station, the near user operates as OW-DF relay node, and far user operates as terminal node B [39]. Moreover, a two-step iteratively training policy is proposed in [67] where the AE frameworks in the first and second phases are trained iteratively, while a two-step

training policy by removing the iterative loop is proposed in [67] with an information-theoretic perspective. Moreover, the focus on AE-based OW-DF relay network was to perform modulation design [37, 39, 36], while the a symbol-wise AE-based differential block coded modulation (d-BCM) design is considered in [36, 67] for transmission rate $R = 4/14$ [bits/channel-reuse] considering $(7, 4)$ Hamming code as baseline error correction code and BPSK modulation, where 14 is because of two-phase transmission of $n = 7$ symbols.

In our recent work [67], we analyzed symbol-wise AE-based two-way DF (TW-DF) relay networks, wherein the terminal nodes A and B communicate with each other employing a TW-DF relay node, thereby increasing the spectral efficiency by 33% (c.f. [22], [56]) in comparison to AE-based OW-DF relay networks [37, 39, 36], [67]. In particular, the first phase has a different AE framework, wherein an NN-based encoder at terminal node A encodes the signal and transmits it to the TW-DF relay node, which decodes the signal using the first NN-based decoder. The second phase has a different AE framework, wherein an NN-based encoder at terminal node B encodes the signal and transmits it to the TW-DF relay node, which decodes the signal using the second NN-based decoder. Thus, the TW-DF relay node employs two separate NN-based decoders, the output of both of which are concatenated and passed as input to the AE in the third phase. The third phase employs an NN-based encoder which encodes both the signals together and broadcasts it to both the terminal nodes A and B, which employ separate NN-based decoders to decode the signal of the other terminal node. We proposed to train the three AE frameworks in an iterative three-step training policy. Moreover, we considered a symbol-wise AE-based block coded modulation (BCM) design for transmission rate $R = 8/21$ [bits/channel-reuse] considering $(7, 4)$ Hamming code as baseline error correction code and BPSK modulation, where 21 is because of three-phase transmission of $n = 7$ symbols and 8 is because both terminal nodes communicate with each other, for $k = 4$ bits.

Furthermore, as detailed in Chapter 4, in practice, relaying systems are compromised by in-phase (I) and quadrature-phase (Q) imbalance (IQI) deteriorating the network performance [19]–[71]. Furthermore, the removal of the IQI requires the channel state information (CSI) knowledge and estimation of the IQI parameters and CSI separately. Thereby leading to increased feedback overhead.

Thus, in contrast to the existing literature, as shown in Table 7.1.1, the major contributions of this chapter are as follows:

- We propose a stacked bit-wise AE-based OW-DF relay network in the presence of IQI at all the nodes. We propose to employ two different AE frameworks in two phases of the transmission, train these two AEs separately and stack them together for predictions. Specifically, we propose a stacked bit-wise AE and *denoising bit-wise AE* frameworks in the first and second phases of transmission, respectively. We

Table 7.1.1: Comparison of proposed Chapter versus state-of-the-art AE literature of DF relay networks [37, 39, 36], [67].

Relay Network	Ref No.	Bit-wise AE	BCM design	d-BCM design	I/Q Imbalance	Rate R
<i>OW-DF</i>	[36]	✗	✗	✓	✗	4/14
	[37]	✗	✗	✗	✗	N/A
	Our previous [67]	✗	✗	✓	✗	4/14
	[39]	✓	✗	✗	✗	N/A
	This Chapter	✓	✓	✓	✓	8/14
<i>TW-DF</i>	Our previous [67]	✗	✓	✗	✗	8/21
	This chapter	✓	✓	✓	✓	16/21

utilize the benefits of soft-decision decoding by the bit-wise AE in the first phase by passing it directly as input to the denoising bit-wise AE in the second phase. Also, we propose a new two-step training scheme for training the proposed stacked AE frameworks.

- We propose a stacked bit-wise point-to-point (P2P) AE-based TW-DF relay network in the presence of IQI at all the nodes. Furthermore, we propose employing a bit-wise AE framework of a P2P network to design a bit-wise AE-based TW-DF relay network. Specifically, we firstly train a bit-wise AE framework (in a single-step training policy) for a P2P network. Secondly, we stack the NN-based encoder-decoder of the trained P2P AE framework to design an AE-based TW-DF relay network. Lastly, we utilize the benefits of conventional bit-wise XOR-based encoding at the relay node and decoding at the terminal nodes to remove each terminal's own bits.
- For both the OW-DF and TW-DF relay networks, we propose AE-based BCM and d-BCM designs such that the need for IQI parameter information is removed, thereby reducing the feedback overhead.
- For both the OW-DF and TW-DF relay networks, we propose a training dataset creation policy such that a single trained AE framework-based BCM and d-BCM designs can generalize well on any levels of testing IQI and signal-to-noise-ratio (SNR).

7.2 System Model

In this section, we detail the conventional DF relay networks, wherein the terminal nodes (A and B) communicate using a DF relay node (R), and each node is impacted by the IQI. Each node has a single antenna, and the direct link between the terminal nodes

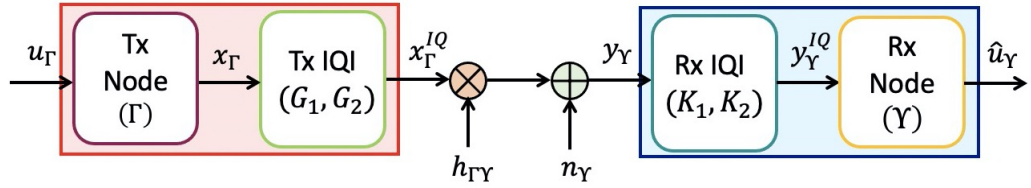


Figure 7.2.1: Signal transmission between the Tx node (Γ) and the Rx node (Υ) with IQI.

is absent because of large-scale path-loss and shadowing effects. The terminal nodes perform block-by-block encoding-decoding, while the signal transmission in each phase takes place as symbol-by-symbol. For ease of explanation, we keep the block length n as unity in this section. Further, the modeling of the IQI remains the same as Section 4.2.1.

7.2.1 Signal Transmission–Reception between any Two Nodes

For sake of brevity, we enlist the steps for signal transmission–reception between a Tx node $\Gamma = \{A, B, R\}$ and a Rx node $\Upsilon = \{A, B, R\}$, where $\Gamma \neq \Upsilon$, as shown in Fig. 7.2.1, to be utilized later on in this section, as below

1. *At the Tx node:* The Tx node Γ intends to transmit $\mathbf{u}_\Gamma \in \{0, 1\}^k$ bits by mapping \mathbf{u}_Γ to a complex baseband symbol $x_\Gamma = \mathcal{M}(\mathbf{u}_\Gamma) \in \mathbb{C}$ using the $\mathcal{M}(\cdot)$ modulation function, such that $\mathbb{E}\{|x_\Gamma|^2\} = 1$. The up-converted signal in presence of Tx IQI becomes x_Γ^{IQ} , given by

$$x_\Gamma^{IQ} = G_1 x_\Gamma + G_2^* x_\Gamma^* \quad (7.1)$$

where $(\cdot)^*$ denotes conjugate operation.

2. *Signal transmission between Tx and Rx node:* Let the signal received at the Rx node Υ be y_Υ , given by

$$y_\Upsilon = \sqrt{P_\Gamma} h_{\Gamma\Upsilon} x_\Gamma^{IQ} + n_\Upsilon \quad (7.2)$$

where P_Γ denotes the transmission power of Tx node Γ , $h_{\Gamma\Upsilon}$ represents the fading channel between the Tx node Γ and Rx node Υ , such that $h_{\Gamma\Upsilon} \sim \mathcal{CN}(0, 1)$, and n_Υ is the additive white Gaussian noise (AWGN) at the Rx node Υ such that $n_\Upsilon \sim \mathcal{CN}(0, \sigma_\Upsilon^2)$.

3. *At the Rx node:* Considering the Rx side IQI effect, we obtain y_{Υ}^{IQ} , given by

$$y_{\Upsilon}^{IQ} = K_1 y_{\Upsilon} + K_2 y_{\Upsilon}^* \quad (7.3)$$

$$\begin{aligned} &= \underbrace{\sqrt{P_{\Gamma}} (K_1 G_1 h_{\Gamma\Upsilon} + K_2 G_2 h_{\Gamma\Upsilon}^*)}_{\text{Desired signal}} x_{\Gamma} + \\ &\quad \underbrace{\sqrt{P_{\Gamma}} (K_1 G_2^* h_{\Gamma\Upsilon} + K_2 G_1^* h_{\Gamma\Upsilon}^*)}_{\text{Self-interference signal}} x_{\Gamma}^* + \underbrace{K_1 n_{\Upsilon} + K_2 n_{\Upsilon}^*}_{\text{Noise}} \\ &= \Lambda(\Gamma, \Upsilon) x_{\Gamma} + \Omega(\Gamma, \Upsilon) x_{\Gamma}^* + \tilde{n}_{\Upsilon}(\Gamma, \Upsilon) \end{aligned} \quad (7.4)$$

where

$$\Lambda(\Gamma, \Upsilon) = \sqrt{P_{\Gamma}} (K_1 G_1 h_{\Gamma\Upsilon} + K_2 G_2 h_{\Gamma\Upsilon}^*) \quad (7.5)$$

$$\Omega(\Gamma, \Upsilon) = \sqrt{P_{\Gamma}} (K_1 G_2^* h_{\Gamma\Upsilon} + K_2 G_1^* h_{\Gamma\Upsilon}^*) \quad (7.6)$$

$$\tilde{n}_{\Upsilon}(\Gamma, \Upsilon) = K_1 n_{\Upsilon} + K_2 n_{\Upsilon}^* \quad (7.7)$$

Remark 25 Please note in (7.4) we can see that the IQI leads to signal distortion, $\Lambda(\Gamma, \Upsilon)x_{\Gamma}$, and causes self-interference, $\Omega(\Gamma, \Upsilon)x_{\Gamma}^*$, thereby deteriorating the network performance.

4. *ZF-based IQI compensation at the Rx node:* Assuming the Rx node (Υ) has the IQI parameters ($G_{(\cdot)}$, $K_{(\cdot)}$) information the ZF-based IQI compensation can be performed to remove the IQI. Wherein, we aim to cancel the self interference term in (7.4) and detect the transmitted signal x_{Γ} . Similar to Chapters 4, 5, this is achieved by concatenating the received signal y_{Υ}^{IQ} and its conjugate y_{Υ}^{IQ*} in a matrix form as follows:

$$\begin{aligned} \begin{bmatrix} y_{\Upsilon}^{IQ} \\ y_{\Upsilon}^{IQ*} \end{bmatrix} &= \begin{bmatrix} \Lambda(\Gamma, \Upsilon) & \Omega(\Gamma, \Upsilon) \\ \Omega(\Gamma, \Upsilon)^* & \Lambda(\Gamma, \Upsilon)^* \end{bmatrix} \begin{bmatrix} x_{\Gamma} \\ x_{\Gamma}^* \end{bmatrix} + \begin{bmatrix} K_1 & K_2 \\ K_2^* & K_1^* \end{bmatrix} \begin{bmatrix} n_{\Upsilon} \\ n_{\Upsilon}^* \end{bmatrix} \\ \mathbf{y}_{\Upsilon}^{IQ} &= \mathbf{A}(\Gamma, \Upsilon) \mathbf{x}_{\Gamma} + \mathbf{B}(\Gamma, \Upsilon) \mathbf{n}_{\Upsilon} \end{aligned} \quad (7.8)$$

We perform ZF-based IQI compensation to get \hat{y}_{Υ}^{IQ} , as

$$\begin{bmatrix} \hat{y}_{\Upsilon}^{IQ} \\ \hat{y}_{\Upsilon}^{IQ*} \end{bmatrix} = (\mathbf{A}(\Gamma, \Upsilon))^{-1} \times \mathbf{y}_{\Upsilon}^{IQ} \quad (7.9)$$

Remark 26 Please note in ZF-based IQI compensation, we assume IQI parameters information, that needs to be estimated separately, thus in the absence of IQI parameter information, i.e. no IQI compensation, Step 4 is omitted and $\hat{y}_{\Upsilon}^{IQ} = y_{\Upsilon}^{IQ}$.

5. *MLD at the Rx node:* Assuming the Rx node Υ has the CSI knowledge ($h_{\Gamma\Upsilon}$) and

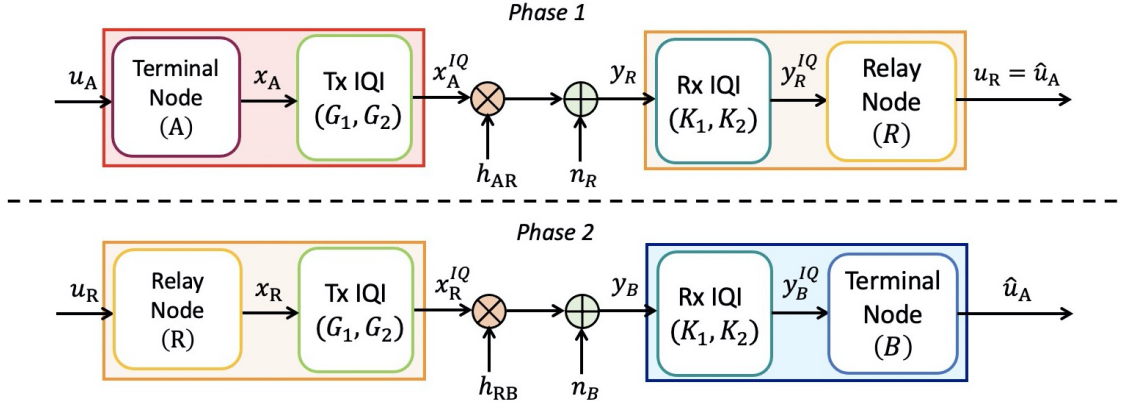


Figure 7.2.2: One-way decode-and-forward relay networks with IQI.

transmission power (P_Γ) information, the Rx node performs optimal MLD as

$$\hat{\mathbf{u}}_\Gamma = \arg \min_{x \in \mathcal{C}} \left\| \hat{y}_\Gamma^{IQ} - \sqrt{P_\Gamma} h_{\Gamma\Upsilon} x \right\|^2 \quad (7.10)$$

where \mathcal{C} denotes all the possible symbols and $\hat{\mathbf{u}}_\Gamma$ is decoded bits of size k intended by the Tx node (Γ).

7.2.2 One-Way Decode-and-Forward Relay Networks with IQI

As shown in Fig. 7.2.2, we consider an OW-DF relay network with IQI at all the nodes, where a terminal node A transmits the signals to the terminal node B with the aid of an OW-DF relay node R, with an effective transmission rate $R = k/2n$ [bits/channel reuse], where 2 is because transmission takes place in two phases.

In the first phase, the terminal node A transmits $\mathbf{u}_A = \{0, 1\}^k$ bits to the relay node R using the signal transmission-reception procedure in Sec. 7.2.1 with $\Gamma = A$ and $\Upsilon = R$, to obtain the decoded k bits $\hat{\mathbf{u}}_A$ at the relay node R. In the second phase, the relay node R re-transmits $\mathbf{u}_R = \hat{\mathbf{u}}_A$ to the terminal node B using the signal transmission-reception procedure in Sec. 7.2.1 with $\Gamma = R$ and $\Upsilon = B$, to obtain the decoded k bits $\hat{\mathbf{u}}_A$ intended by the terminal node A.

7.2.3 Two-Way Decode-and-Forward Relay Networks with IQI

As shown in Fig. 7.2.3, we consider a TW-DF relay network with IQI at all the nodes, where a terminal node A communicates with the terminal node B (and vice-versa) with the aid of a TW-DF relay node R, with an effective transmission rate $R = 2k/3n$ [bits/channel reuse], where 2 and 3 are because both the terminal communicate with each other and transmission takes place in three phases, respectively.

In the first phase, the terminal node A transmits $\mathbf{u}_A = \{0, 1\}^k$ bits to the relay node R

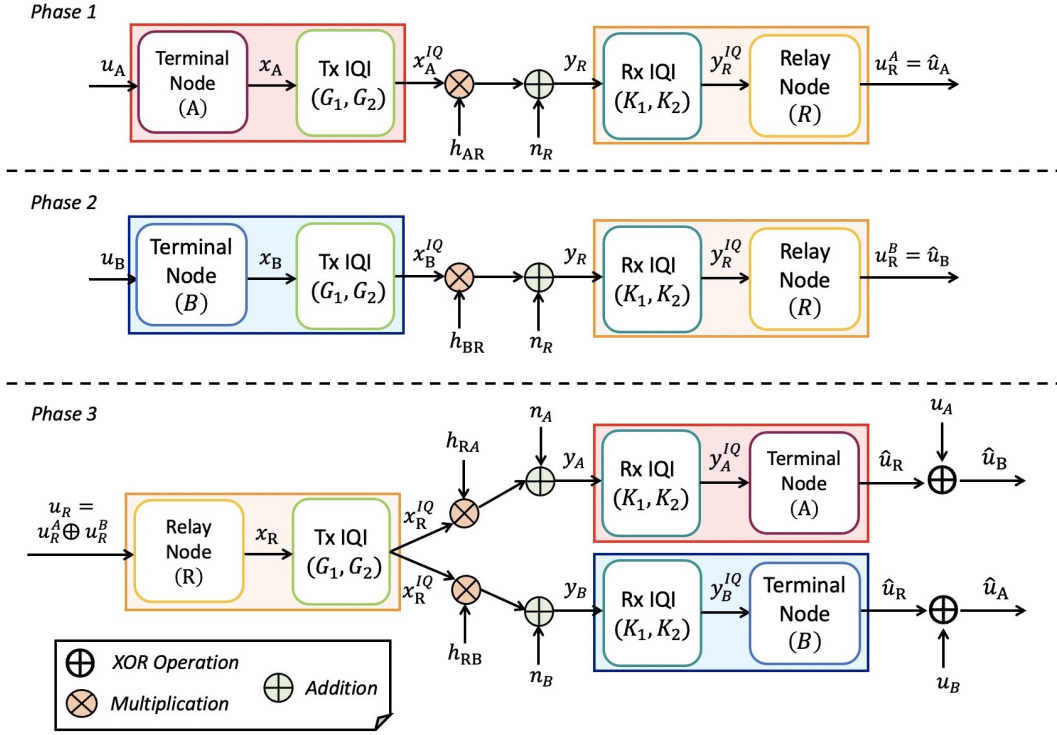


Figure 7.2.3: Two-way decode-and-forward relay networks with IQI.

using the signal transmission-reception procedure in Sec. 7.2.1 with $\Gamma = A$ and $\Upsilon = R$, to obtain the decoded k bits $\mathbf{u}_R^A = \hat{\mathbf{u}}_A$ at the relay node R. In the second phase, the terminal node B transmits $\mathbf{u}_B = \{0, 1\}^k$ bits to the relay node R using the signal transmission-reception procedure in Sec. 7.2.1 with $\Gamma = B$ and $\Upsilon = R$, to obtain the decoded k bits $\mathbf{u}_R^B = \hat{\mathbf{u}}_B$ at the relay node R. For the third phase, we detail the signal transmission-reception procedure using Steps 1-5 mentioned in Sec. 7.2.1, as follows:

- The relay node R performs bit-wise XOR operation on the decoded bits in the first and second phases, as follows:

$$\mathbf{u}_R = \mathbf{u}_R^A \oplus \mathbf{u}_R^B \quad (7.11)$$

where \oplus denotes the bit-wise XOR operation.

- Similar to Step 1 (in Sec. 7.2.1), the relay node obtains the up-converted signal x_R^{IQ} by keeping $\Gamma = R$.
- Then similar to Steps 2 and 3 (in Sec. 7.2.1), the relay node broadcasts the signal to the terminal nodes A and B by keeping $\Gamma = R, \Upsilon = A$ and $\Gamma = R, \Upsilon = B$, respectively.
- Now using Steps 4 and 5 (in Sec. 7.2.1) the terminal nodes A and B decode the k bits $\hat{\mathbf{u}}_R$ transmitted by the relay node by keeping $\Gamma = R, \Upsilon = A$ and $\Gamma = R, \Upsilon = B$,

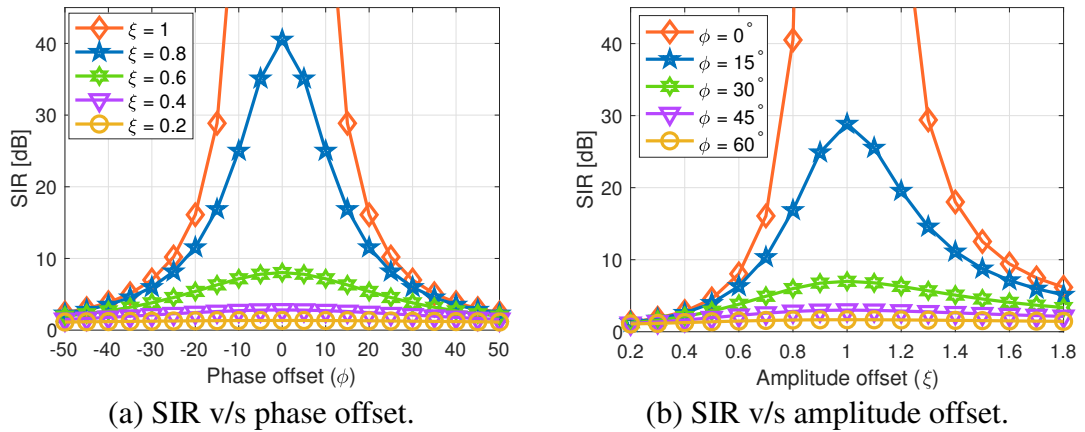


Figure 7.2.4: Impact of IQI on the SIR (in dB) in OW-DF and TW-DF relay networks.

respectively.

- Since each terminal node has its own information bits (i.e. terminal node A and B has information about \mathbf{u}_A and \mathbf{u}_B , respectively), it can now perform bit-wise XOR on the decoded bits and its own bits to decode the other terminal node's bits (i.e. terminal node A and B decodes $\hat{\mathbf{u}}_B$ and $\hat{\mathbf{u}}_A$, respectively), as follows:

$$\text{At terminal node A : } \quad \hat{\mathbf{u}}_B = \hat{\mathbf{u}}_R \oplus \mathbf{u}_A \quad (7.12)$$

$$\text{At terminal node B : } \quad \hat{\mathbf{u}}_A = \hat{\mathbf{u}}_R \oplus \mathbf{u}_B \quad (7.13)$$

7.2.4 Impact of IQI on OW-DF and TW-DF Relay Networks

We analyze the destructive effects of the IQI on the OW-DF and TW-DF relay networks. For simplicity, we assume the transmission power of each node $P_\Gamma = 1$ where $\Gamma = \{A, B, R\}$ and we have equal Tx and Rx side IQI at each node, $\xi_T = \xi_R = \xi$ and $\phi_T = \phi_R = \phi$, throughout this Chapter.

Considering there are no noise terms $n_\Gamma = 0$, we can determine the signal-to-interference-ratio (SIR in dB) for the OW-DF and TW-DF relay networks¹ using (7.4) as follows:

$$\text{SIR} = \frac{\mathbb{E} \{ |\Lambda(\Gamma, \Upsilon) x_\Gamma|^2 \}}{\mathbb{E} \{ |\Omega(\Gamma, \Upsilon) x_\Gamma^*|^2 \}} = \frac{|K_1|^2 |G_1|^2 + |K_2|^2 |G_2|^2}{|K_1|^2 |G_2|^2 + |K_2|^2 |G_1|^2} \quad (7.14)$$

In Fig. 7.2.4, we analyze the impact of varying phase and amplitude offsets on the SIR in (7.14). In the ideal scenario with no IQI, the SIR becomes *infinity*. Whereas, even a small phase offset $\phi = 15^\circ$ or amplitude offset $\phi = 0.8$ can deteriorate the SIR to 30 dB or 40 dB, respectively. Furthermore, in the scenarios with larger phase offsets $\phi \geq 30^\circ$ or amplitude offsets $\xi \leq 0.6$, the SIR remains less than 8 dB. Thus, even smaller IQI can

¹The SIR (in dB) remains the same in each hop for OW-DF and TW-DF relay networks because we consider same Tx and Rx side IQI at each node and the relay node performs in DF mode.

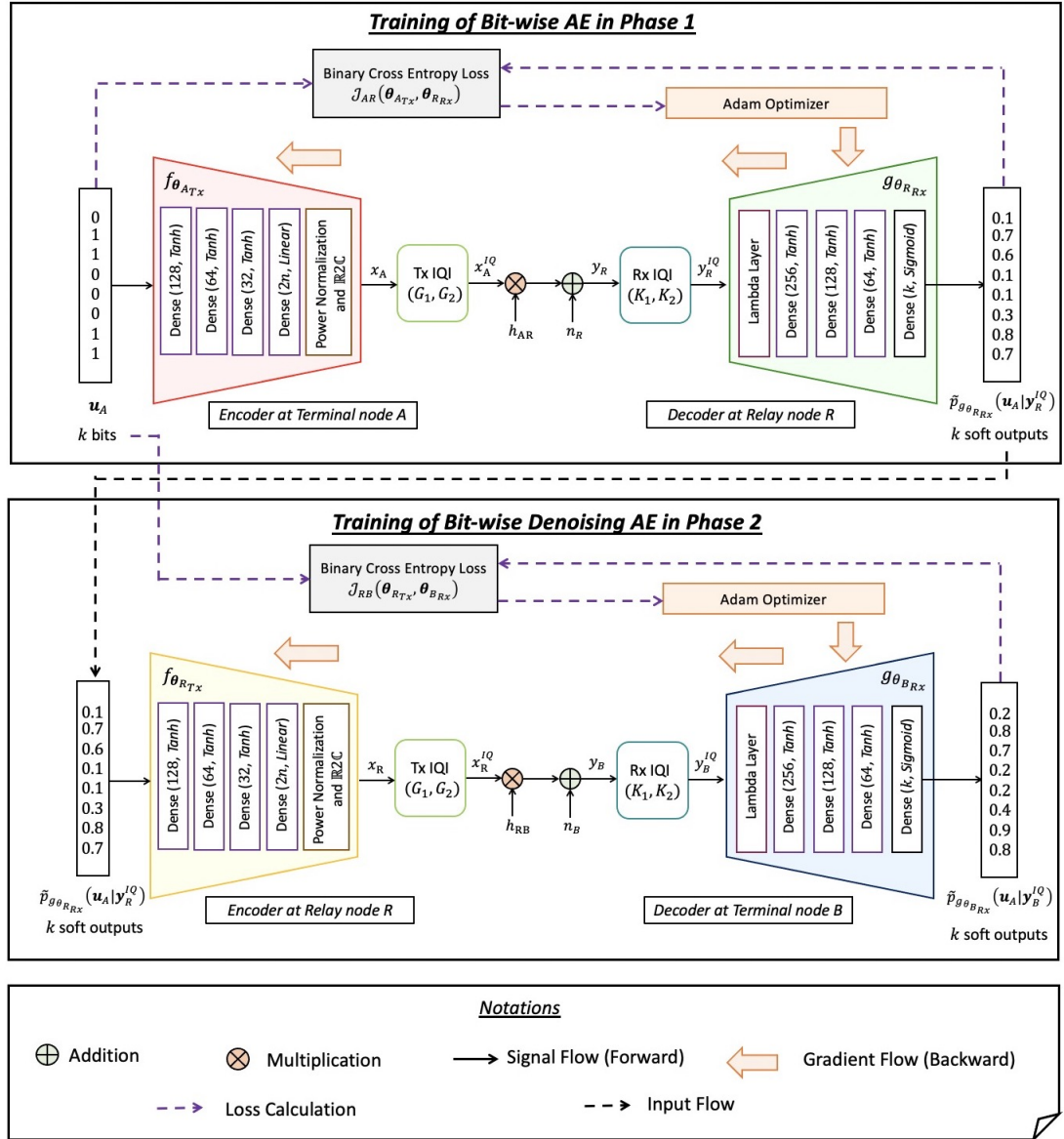


Figure 7.3.1: Training of proposed stacked bit-wise AE and bit-wise denoising AE frameworks for OW-DF relay networks with IQI.

deteriorate the SIR significantly. Hence it becomes pivotal to develop novel techniques for OW-DF and TW-DF relay networks that can help in the successful transmission of signals in the presence of IQI under such low SIR regimes.

7.3 Proposed Stacked Bit-wise AE and Denoising AE-based OW-DF Relay Networks with IQI

In this section, we propose stacked AEs-based BCM and d-BCM designs for the OW-DF relay network with IQI at all the nodes.

As detailed in Sec. 7.2, we assume the direct link between the terminal nodes is absent,

and the relay node operates in DF mode. Thus, the two-phase transmissions in an OW-DF relay network can be considered as two separate P2P transmissions. Hence, we consider two different AE frameworks in each phase. In this chapter, we propose to employ the bit-wise AE frameworks that decode k soft probabilistic outputs at their NN decoders. Specifically, in the first phase, we employ a bit-wise AE framework with its NN encoder at the terminal node A and its NN decoder at the relay node R. The bit-wise AE is trained in an end-to-end manner, as shown in Fig. 7.3.1: ‘Training of the bit-wise AE in Phase 1’, with its input as k bits and output as k soft probabilities.

Now, for the first time, we introduce the concept of bit-wise denoising AE framework, defined as follows:

Definition 22 (Bit-wise denoising Autoencoder) *A bit-wise denoising AE is a bit-wise AE framework, with the difference that the input at the NN encoder is the soft probabilistic values lying between $[0, 1]$ instead of bits $\{0, 1\}$.*

Remark 27 *Please note that the bit-wise denoising AE framework also employs a Sigmoid activation in the last layer of the NN decoder to retrieve soft probabilistic outputs and is optimized by minimizing the binary CE loss.*

In the second phase, we propose to employ a bit-wise denoising AE framework with its NN encoder at the relay node R and its NN decoder at the terminal node B. This is because the NN decoder (of bit-wise AE) at the relay node R (in the first phase) produces soft probabilistic outputs, which can be directly fed as an input to the NN encoder of the bit-wise denoising AE framework.

Now, we propose a new end-to-end training for the proposed bit-wise denoising AE, as shown in Fig. 7.3.1 – ‘Training of the bit-wise denoising AE in Phase 2’. Specifically, we perform end-to-end training between the k input bits of the NN encoder of the bit-wise AE in the first phase transmission and k soft probabilistic outputs of the NN decoder of the bit-wise denoising AE framework in the second phase transmission.

A direct advantage of employing the bit-wise denoising AE is that we remove the hard decision decoding (HDD) on the soft outputs of the NN decoder of the bit-wise AE framework in the first phase. This is because the probability of decoding the bits erroneously is highest for the soft probabilistic outputs lying close to the HDD threshold due to the ambiguity in making the decisions. Suppose a soft probability lies close to the HDD threshold, then the HDD-based bit results can not be trusted completely. Furthermore, in a DF relaying network where the output of the NN decoder of the bit-wise AE in the first phase is fed into the NN encoder of the bit-wise denoising AE in the second phase, the chances of error propagation increase with HDD-based bit results because of ambiguous decoding near HDD threshold. Instead, if we directly utilize the soft probabilistic outputs of the first phase’s bit-wise AE’s NN decoder as the input to the second phase’s bit-wise

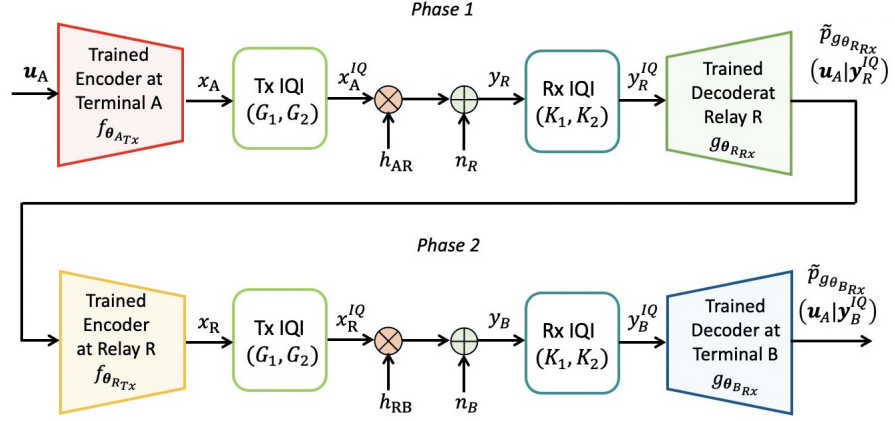


Figure 7.3.2: Testing of proposed stacked bit-wise AE and bit-wise denoising AE frameworks for OW-DF relay networks with IQI.

denoising AE and employ the aforementioned technique to train the bit-wise denoising AE framework. Then the bit-wise denoising AE can learn to decode the soft probabilities lying close to the HDD threshold correctly.

Once both the bit-wise AE and bit-wise denoising AE frameworks are trained, we propose to stack these trained AE frameworks for the two-phase transmissions to form a stacked AEs-based OW-DF relay network, as shown in Fig. 7.3.2.

7.3.1 Designing of the Bit-wise AE for Phase 1

In this chapter, we utilize L fully-connected (dense) layers, wherein the output of the $l^{\text{th}} \in \{1, \dots, L\}$ dense layer $\omega_l \in \mathbb{R}^{\delta_l}$ can be given as (2.2). It is also important to note the Remark 15 here for understanding the notations.

The terminal node A takes k bits $\mathbf{u}_A \in \{0, 1\}^k$ as input and performs block-by-block encoding by mapping \mathbf{u}_A to n complex baseband symbols $\mathbf{x}_A \in \mathbb{C}^n$ using the mapping function $f_{\theta_{ATx}}(\mathbf{u}_A, \mathbf{x}_A)$, where θ_{ATx} is the weight and bias terms of constituent M dense layers, similar to detailed in *Definition 6*, and given by

$$f_{\theta_{ATx}}(\mathbf{u}_A, \mathbf{x}_A) = \mathbf{P}_N(\sigma_M(\mathbf{W}_M \sigma_{M-1}(\mathbf{W}_{M-1} \sigma_{M-2}(\dots \sigma_1(\mathbf{W}_1 \mathbf{u}_A + \mathbf{b}_1) \dots) + \mathbf{b}_{M-1}) + \mathbf{b}_M)) \quad (7.15)$$

wherein, \mathbf{P}_N denotes the power normalization layer, detailed in *Definition 7*, that mandates $\|\mathbf{x}_A\|_2^2 = n$.

Then, the symbol-by-symbol transmission reception takes place between the terminal node A and relay node R. In particular, first the symbol is up-converted in presence of Tx IQI to become x_A^{IQ} using (7.1) with $\Gamma = A$. Then, the signal received by the relay node R becomes y_R using (7.2) with $\Gamma = A$ and $\Upsilon = R$. Lastly, by considering the Rx side IQI,

the received signal y_R at the relay node R becomes y_R^{IQ} using (7.4) with $\Upsilon = R$.

Once the NN decoder at the relay node R receives these n symbols, given by $\mathbf{y}_R^{IQ} \in \mathbb{C}^n$, it performs block-by-block decoding by de-mapping \mathbf{y}_R^{IQ} to k soft probabilistic outputs $\tilde{p}_{g\theta_{R_{Rx}}}(\mathbf{u}_A|\mathbf{y}_R^{IQ}) \in [0, 1]^k$, as detailed in *Definition 8*, using the de-mapping function $g_{\theta_{R_{Rx}}}(\mathbf{y}_R^{IQ}, \tilde{p}_{g\theta_{R_{Rx}}}(\mathbf{u}_A|\mathbf{y}_R^{IQ}))$, where $\theta_{R_{Rx}}$ is weight and bias terms of constituent N dense layers, as detailed in *Definition 9*, given by

$$g_{\theta_{R_{Rx}}}(\mathbf{y}_R^{IQ}, \tilde{p}_{g\theta_{R_{Rx}}}(\mathbf{u}_A|\mathbf{y}_R^{IQ})) = \sigma_N(\mathbf{W}_N \sigma_{N-1}(\dots \sigma_1(\mathbf{W}_1 \mathbf{L}_L(\mathbf{y}_R^{IQ}) + \mathbf{b}_1) \dots) + \mathbf{b}_N) \quad (7.16)$$

where \mathbf{L}_L denotes the Lambda layer detailed in *Definition 10*.

7.3.2 Designing of the Bit-wise Denoising AE for Phase 2

The relay node R takes the k soft probabilistic outputs $\tilde{p}_{g\theta_{R_{Rx}}}(\mathbf{u}_A|\mathbf{y}_R^{IQ}) \in [0, 1]^k$ obtained from the NN decoder of trained bit-wise AE (in first phase) as input and performs block-by-block encoding by mapping $\tilde{p}_{g\theta_{R_{Rx}}}(\mathbf{u}_A|\mathbf{y}_R^{IQ})$ to n complex baseband symbols $\mathbf{x}_R \in \mathbb{C}^n$ using mapping function $f_{\theta_{R_{Tx}}}(\tilde{p}_{g\theta_{R_{Rx}}}(\mathbf{u}_A|\mathbf{y}_R^{IQ}), \mathbf{x}_R)$, where $\theta_{R_{Tx}}$ is weight and bias terms of constituent M dense layers given by

$$f_{\theta_{R_{Tx}}}(\tilde{p}_{g\theta_{R_{Rx}}}(\mathbf{u}_A|\mathbf{y}_R^{IQ}), \mathbf{x}_R) = \mathbf{P}_N(\sigma_M(\mathbf{W}_M \sigma_{M-1}(\dots \sigma_1(\mathbf{W}_1 \tilde{p}_{g\theta_{R_{Rx}}}(\mathbf{u}_A|\mathbf{y}_R^{IQ}) + \mathbf{b}_1) \dots) + \mathbf{b}_M)) \quad (7.17)$$

where power normalization layer (\mathbf{P}_N) ensures $\|\mathbf{x}_R\|_2^2 = n$. Then, the symbol-by-symbol transmission reception takes place between the relay node R and terminal node B. In particular, first the symbol is up-converted in presence of Tx IQI to become x_R^{IQ} using (7.1) with $\Gamma = R$. Then, the signal received by the terminal node B becomes y_B using (7.2) with $\Gamma = R$ and $\Upsilon = B$. Lastly, by considering the Rx side IQI, the received signal y_B at the terminal node B becomes y_B^{IQ} using (7.4) with $\Upsilon = B$.

Once NN decoder at terminal node B receives n symbols, given by $\mathbf{y}_B^{IQ} \in \mathbb{C}^n$, it performs block-by-block decoding by de-mapping \mathbf{y}_B^{IQ} to k soft probabilistic outputs $\tilde{p}_{g\theta_{B_{Rx}}}(s_A^m|\mathbf{y}_B^{IQ}) \in [0, 1]$, for all m , as detailed in *Definition 8*, using de-mapping function $g_{\theta_{B_{Rx}}}(\mathbf{y}_B^{IQ}, \tilde{p}_{g\theta_{B_{Rx}}}(\mathbf{u}_A|\mathbf{y}_B^{IQ}))$, where $\theta_{B_{Rx}}$ is weight and bias terms of constituent N dense layers, given by

$$g_{\theta_{B_{Rx}}}(\mathbf{y}_B^{IQ}, \tilde{p}_{g\theta_{B_{Rx}}}(\mathbf{u}_A|\mathbf{y}_B^{IQ})) = \sigma_M(\mathbf{W}_M \sigma_{M-1}(\dots \sigma_1(\mathbf{W}_1 \mathbf{L}_L(\mathbf{y}_B^{IQ}) + \mathbf{b}_1) \dots) + \mathbf{b}_M) \quad (7.18)$$

Remark 28 Please note, as shown in Fig. 7.3.1 – ‘Training of the bit-wise denoising AE in Phase 2’, we propose end-to-end training between the input bits of the NN encoder of the bit-wise AE (in the first phase) and soft outputs of NN decoder of the bit-wise denoising AE. Thus, NN decoder of bit-wise denoising AE learns the distribution $\tilde{p}_{g_{\theta_{B_{Rx}}}}(\mathbf{u}_A | \mathbf{y}_B^{IQ})$, learning the soft probabilistic outputs for input bits at terminal node A. If end-to-end training between the input-output of the bit-wise denoising AE have been performed, then the NN decoder of bit-wise denoising AE would have learnt the distribution $\tilde{p}_{g_{\theta_{B_{Rx}}}(\tilde{p}_{g_{\theta_{R_{Rx}}}}(\mathbf{u}_A | \mathbf{y}_R^{IQ}) | \mathbf{y}_B^{IQ})}$, learning the soft probabilistic outputs of the NN decoder of bit-wise AE in first phase. Thereby showing the merits of proposed training of bit-wise denoising AE framework (discussed in Sec. 7.3.5).

7.3.3 Proposed AE-based BCM and d-BCM Designs

Similar to Chapter 2, we propose block-by-block encoding and decoding at the NN encoders and decoders, respectively, while the proposed AE-based BCM with CSI knowledge and d-BCM without CSI is determined on the basis of the designed Lambda layer (\mathbf{L}_L) in the NN decoders, detailed as follows:

- *BCM* – Herein, we assume the CSI knowledge and perform channel equalization in Lambda layer for the $h_{\Gamma\Upsilon}$ between the transmitter node Γ and receiver node Υ .
- *d-BCM* – Herein, we assume we do not have the CSI knowledge and propose to employ a radio transformer network (RTN). The RTN has been widely employed to estimate the CSI knowledge [12]; however, similar to Chapter 4, we propose an RTN that also helps in removing the IQI from the received signal at the receiver (decoder) node Υ .

Please note, unlike the conventional networks considering ZF-based IQI compensation (as discussed in Step 4 in Sec. 7.2.1), with the knowledge of IQI parameters, our proposed AE framework do not require the IQI parameter information in BCM or d-BCM designs. Thus, removing the feedback overhead for IQI estimation.

7.3.4 Loss Optimization

The proposed bit-wise AE and bit-wise denoising AE frameworks are optimized by solving the multi-label binary classification problem, by minimizing the binary CE losses $\mathcal{L}_{AR}(\mathbf{u}_A, \tilde{p}_{g_{\theta_{R_{Rx}}}(\mathbf{u}_A | \mathbf{y}_R^{IQ})})$ and $\mathcal{L}_{RB}(\mathbf{u}_A, \tilde{p}_{g_{\theta_{B_{Rx}}}(\mathbf{u}_A | \mathbf{y}_B^{IQ})})$ for first and second transmission phases, respectively, each of which is defined similar to the *Definition 13*.

Table 7.3.1: NN architectures at different nodes for any of the proposed AE frameworks.

NN Encoder		NN Decoder		RTN in Lambda layer	
Neurons	Remark	Neurons	Remark	Neurons	Remark
$\delta_1 = k$	Input	$\delta_1 = 2n$	Input	$\delta_1 = 2n$	Input
$\delta_2 = 128$	$\sigma_1 = \text{Tanh}$	$\delta_2 = 2n$	Lambda layer (\mathbf{L}_L)	$\delta_2 = 128$	$\sigma_1 = \text{Tanh}$
$\delta_3 = 64$	$\sigma_2 = \text{Tanh}$	$\delta_3 = 256$	$\sigma_1 = \text{Tanh}$	$\delta_3 = 32$	$\sigma_2 = \text{Tanh}$
$\delta_4 = 32$	$\sigma_3 = \text{Tanh}$	$\delta_4 = 128$	$\sigma_2 = \text{Tanh}$	$\delta_4 = 2$	$\sigma_3 = \text{Linear}$
$\delta_5 = 2n$	$\sigma_3 = \text{Linear}$	$\delta_5 = 64$	$\sigma_2 = \text{Tanh}$	$\delta_5 = 2n$	Equalize $2n$ Input by estimated channel 2 output of previous layer
$\delta_6 = 2n$	Power normalization (\mathbf{P}_N)	$\delta_6 = k$	$\sigma_3 = \text{Sigmoid}$		
$\delta_7 = 2n$	Output	$\delta_7 = k$	Soft output		

7.3.5 Training Procedure

Following the procedure in Section 2.5.1, we first train the bit-wise AE in the first phase transmission by minimizing the binary CE loss $\mathcal{L}_{AR}(\mathbf{u}_A, \tilde{p}_{g\theta_{R_{Rx}}}(\mathbf{u}_A|\mathbf{y}_R^{IQ}))$ also represented as $\mathcal{J}_{AR}(\theta_{A_{Tx}}, \theta_{R_{Rx}})$ to obtain the trained parameters $\theta_{A_{Tx}}, \theta_{R_{Rx}}$ of the bit-wise AE. For the trained bit-wise AE (in first phase) we perform predictions on the training set to obtain soft probabilistic outputs $\tilde{p}_{g\theta_{R_{Rx}}}(s_A^m|\mathbf{y}_R^{IQ})$, which are passed as input to the NN encoder of the bit-wise denoising AE (in second phase). Then we train the bit-wise denoising AE by minimizing the binary CE loss $\mathcal{L}_{RB}(\mathbf{u}_A, \tilde{p}_{g\theta_{B_{Rx}}}(\mathbf{u}_A|\mathbf{y}_B^{IQ}))$ also represented as $\mathcal{J}_{RB}(\theta_{R_{Tx}}, \theta_{B_{Rx}})$ to obtain the trained parameters $\theta_{R_{Tx}}, \theta_{B_{Rx}}$ of the bit-wise denoising AE.

7.3.6 NN Architecture

We create a single NN architecture for the encoder, decoder, and RTN to be generalizable for all the proposed AE frameworks, as summarized in Table 7.3.1. We implement all the NNs in Keras [43] with TensorFlow [44] as backend.

7.3.7 Training and Testing Dataset Creation

As detailed in Sec. 7.2.4, we assume $P_\Gamma = 1$, $\Gamma = \{A, B, R\}$ and $\xi_T = \xi_R = \xi$ and $\phi_T = \phi_R = \phi$. We now detail the training dataset creation policy – The E_b/N_0 are taken from the set $\mathcal{S} = \{3, 8, 13, 23, 33, 43, 53, 63\}$ dB, the phase offsets ϕ are taken from the set $\mathcal{P} = \{25^\circ, 35^\circ, 40^\circ, 45^\circ\}$ and the amplitude offsets are taken from the set $\mathcal{A} = \{0.4, 0.5, 0.6, 0.7\}$. In previous Chapters 4, 5, we utilized a separate training dataset for each phase or amplitude offset, overcoming that drawback, in this chapter we create a single training dataset, where we make $Q_{\text{train}} = 2^{k+2}$ blocks of data for each combinations of E_b/N_0 (in dB), phase offsets and amplitude offsets from the sets $[\mathcal{S}, \mathcal{P}, \mathcal{A}]$.

We train a single stacked AE framework on the training dataset and test the trained AE framework for $Q_{\text{test}} = 10^5$ blocks of unseen data for varying E_b/N_0 and IQI levels.

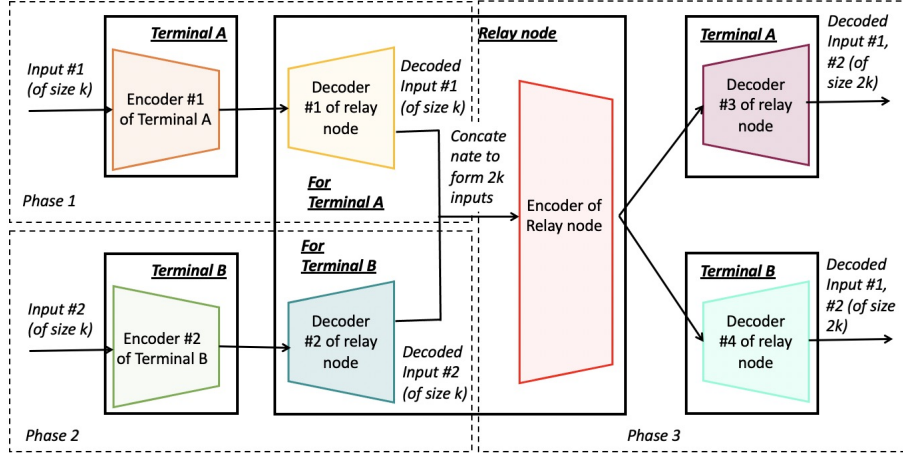


Figure 7.3.3: Separate AE frameworks for each transmission phase in the TW-DF relay networks.

Please note a direct advantage of proposed training is that we create a single training dataset such that a single trained stacked AE framework can generalize well for varying levels of testing E_b/N_0 , phase and amplitude offset (IQI). Also, the training set creation policy remains the same for both the OW-DF and TW-DF AE frameworks for generalizability.

7.3.8 Hyper-Parameter Settings

For all the AE frameworks, we employ SGD with Adam [42] optimizer and initialize the weights using Glorot initializer [45]. We keep the batch size $B = 6000$. We utilize the step-decay method [41] to update the learning rate. Specifically, we start the training with an initial learning rate $\tau_0 = 0.002$ for the first epoch and then drop the learning rate by $\eta = 0.5$ after every $D_E = 25$ epochs (we constraint the minimum learning rate to be $\tau_{\min} = 10^{-5}$). We utilize early stopping [46] to stop the training of the AE if no significant update is made to the validation accuracy during the training procedure. In particular, we keep the minimum update in validation accuracy as 10^{-4} and patience to stop as 17.

7.3.9 Testing or Prediction Phase

During testing, we stack the trained bit-wise AE and bit-wise denoising AE to form the OW-DF relay networks, as shown in Fig. 7.3.2. For the trained bit-wise AE, we perform predictions on the testing set to obtain soft probabilistic outputs $\tilde{p}_{g_{\theta_{R_{Rx}}}}(\mathbf{u}_A | \mathbf{y}_R^{IQ})$, that are passed as input to the bit-wise denoising AE, where we again perform predictions on the testing set to obtain soft probabilistic outputs $\tilde{p}_{g_{\theta_{B_{Rx}}}}(\mathbf{u}_A | \mathbf{y}_B^{IQ})$. Similar to Section 2.5.2, we then perform hard decision decoding by keeping a threshold of 0.5 to obtain the decoded bits at the terminal node B.

7.4 Proposed Stacked P2P AE-based TW-DF Relay Networks with IQI

In this section, we propose a stacked bit-wise P2P AE-based BCM and d-BCM designs for the TW-DF relay network with IQI at all the nodes.

In our previous work [67], a separate symbol-wise AE is trained for each of the three-phase transmissions in the TW-DF relay networks (as detailed in Sec. 7.1). Directly, employing three bit-wise AEs for each transmission phase, as shown in Fig. 7.3.3, can help in improving the performance over [67]. However, we encounter two main problems of employing separate AEs in each phase, as detailed below, even with the bit-wise AE frameworks.

Problem 1: As detailed in Sec. 7.2, in conventional scenarios' third phase transmission, the TW-DF relay node takes advantage of the terminal nodes knowing their information bits by additionally performing bit-wise XOR-based encoding and decoding at the relay and terminal nodes, respectively. However, in Fig. 7.3.3, the NN encoder at the relay node encodes the concatenated signal of both terminal nodes' decoded in the first two phases. Thus the NN decoder at each terminal node ends up decoding the signal for both the terminal nodes. As a result, the NN encoder at relay encodes $2k$ bits in n symbols, while the NN decoders at each terminal node decode $2k$ bits from the n received symbols. Thus, effective transmission rate in third phase becomes $R = 2k/n$ [bits/channel-reuse].

Solution 1: We can resolve the *Problem 1* by additionally performing the bit-wise XOR-based encoding-decoding similar to the conventional scenario. Thus, the NN encoder at the relay node encodes k bits to n symbols. The NN decoders at each terminal node decode k bits from n symbols. Hence, the effective transmission rate in the third phase becomes $R = k/n$.

Advantage 1: We improve the transmission rate in the third phase by a factor of two; thus, smaller NN architectures can be employed and it becomes easier to decode signals.

Problem 2: In Fig. 7.3.3, we need to optimize seven different NN architectures of encoder and decoder pairs for three AEs in three transmission phases. Also, as proposed in [67], we need to perform iterative training to optimize these three AEs.

Solution 2: We can resolve the *Problem 2* by training a single bit-wise P2P AE and re-utilizing it to form a stacked AE-based TW-DF relay network. Specifically, as detailed in Sec. 7.2, we assume the direct link between the terminal nodes is absent, and the relay node operates in DF mode. Thus, the three-phase transmissions in a TW-DF relay network can be considered as four separate P2P transmissions, where the first two P2P transmissions are in the first two phases. In comparison, the third phase's multi-user downlink transmission can be considered as two simultaneous P2P transmissions between the relay and terminal nodes.

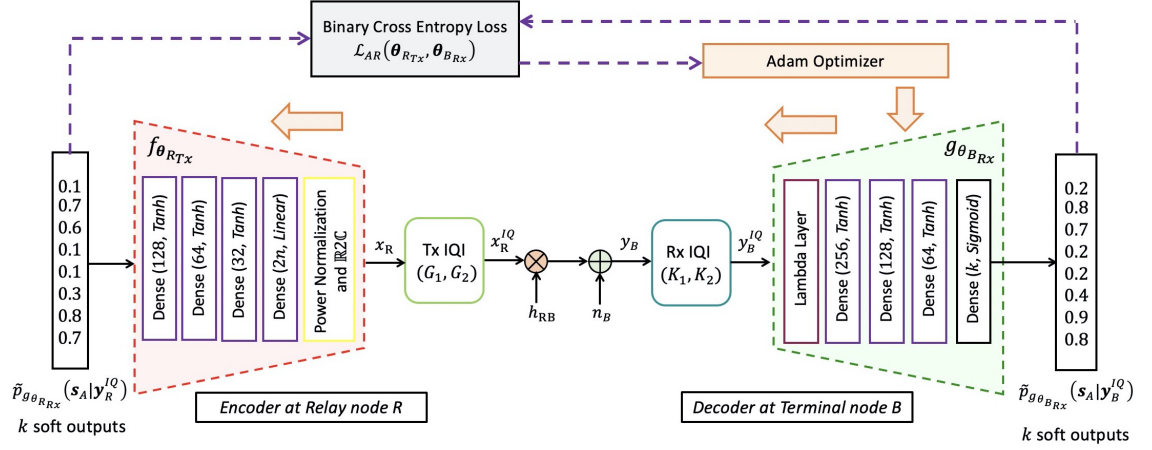


Figure 7.4.1: Training of proposed bit-wise P2P AE framework with IQI.

Advantage 2: We remove the necessity of training seven NN architectures iteratively to only one-time learning of P2P AE.

The aforementioned discussion motivates us to train a single bit-wise AE framework for the P2P network that takes k bits as input at the NN encoder and decodes k soft probabilistic outputs at the NN decoder as shown in Fig. 7.4.1. Then, we propose to re-utilize the trained NN encoder and decoder pair of P2P bit-wise AE in all the nodes (A, B, R) for encoding and decoding purposes in a TW-DF relay network, as depicted in Fig. 7.4.2, and employ the below steps for three-phase transmissions, detailed as follows:

- In the first and second transmission phases, the NN encoder at terminal nodes A and B encodes the signal and transmits it to the relay node's NN decoder that obtains soft probabilistic outputs, which are converted to \mathbf{u}_R^A and \mathbf{u}_R^B bits by hard decision decoding, respectively.
- In the third transmission phase, we propose to employ bit-wise XOR-based encoding of the signal as $\mathbf{u}_R = \mathbf{u}_R^A \oplus \mathbf{u}_R^B$, these encoded bits are passed through the NN encoder at the relay node R. While each terminal node decodes the transmitted bits \mathbf{u}_R using its NN decoder and applying hard decision decoding. Then, each terminal node performs a bit-wise XOR operation with its bits to obtain the decoded bits of the other terminal node.

7.4.1 Designing of Bit-wise P2P AE Framework

We design the bit-wise P2P AE comprising a source node S and destination node D with IQI at both the Tx and Rx sides. The source node S takes k bits $\mathbf{u}_s \in \{0, 1\}^k$ as input and performs block-by-block encoding by mapping \mathbf{u}_s to n complex baseband symbols $\mathbf{x}_s \in \mathbb{C}^n$ using the mapping function $f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s)$, where θ_s is the weight and bias terms of constituent M dense layers, detailed as Definition 6 and given by (2.3).

Then, the symbol-by-symbol transmission reception takes place between the source node S and destination node D. In particular, first the symbol is up-converted in presence of Tx IQI to become x_s^{IQ} using (7.1) with $\Gamma = s$. Then, the signal received by the destination node D becomes y_d using (7.2) with $\Gamma = s$ and $\Upsilon = d$. Lastly, by considering the Rx side IQI, the received signal y_d at the destination node D becomes y_d^{IQ} using (7.4) with $\Upsilon = d$.

Once the NN decoder at the destination node D receives n symbols, given by $\mathbf{y}_d^{IQ} \in \mathbb{C}^n$, it performs block-by-block decoding by de-mapping \mathbf{y}_d^{IQ} to k soft probabilistic outputs $\tilde{p}_{g_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d^{IQ}) \in [0, 1]^k$, using the de-mapping function $g_{\theta_d}(\mathbf{y}_d^{IQ}, \tilde{p}_{g_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d^{IQ}))$, where θ_d is weight and bias terms of constituent N dense layers, given by

$$g_{\theta_d}(\mathbf{y}_d^{IQ}, \tilde{p}_{g_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d^{IQ})) = \sigma_M(\mathbf{W}_N \sigma_{N-1}(\dots \sigma_1(\mathbf{W}_1 \mathbf{L}_L(\mathbf{y}_d^{IQ}) + \mathbf{b}_1) \dots) + \mathbf{b}_N) \quad (7.19)$$

Please note herein, similar to Sec. 7.3.3 for the OW-DF relay networks, we utilize the same methodology for BCM and d-BCM design in the Lambda layer (\mathbf{L}_L) of the NN decoders. Specifically, we employ channel equalization in BCM designs and NN-based RTN in d-BCM designs. Also, unlike conventional scenarios, we do not use IQI parameter information, reducing the feedback overhead for IQI estimation.

7.4.2 Loss Optimization and Training Procedure

Since we propose a bit-wise P2P AE, the process of loss optimization and training remains the same as the P2P networks proposed in Section 2.5.1.

Please note the following for the proposed bit-wise P2P AE – (1) the NN architecture employed in the bit-wise P2P AE remains the same as Sec. 7.3.6, (2) the process of creating training and testing dataset remains the same as Sec. 7.3.7, and (3) the hyper-parameter settings for the training the bit-wise P2P AE also remains the same as Sec. 7.3.8. Please note herein, we also have the following advantage, we create a single training dataset such that a single trained bit-wise P2P AE framework, when employed in a stacked form to create a TW-DF relay network, can generalize well for varying levels of testing E_b/N_0 and IQI levels.

7.4.3 Testing Phase

We construct an AE-based TW-DF relay network by employing the trained NN encoder and decoder of the bit-wise P2P AE framework at all the nodes (A, B, R) in the TW-DF relay network. For predictions, we follow the below steps:

- In the first phase, the terminal node A takes the k bits $\mathbf{u}_A \in \{0, 1\}^k$ as input and

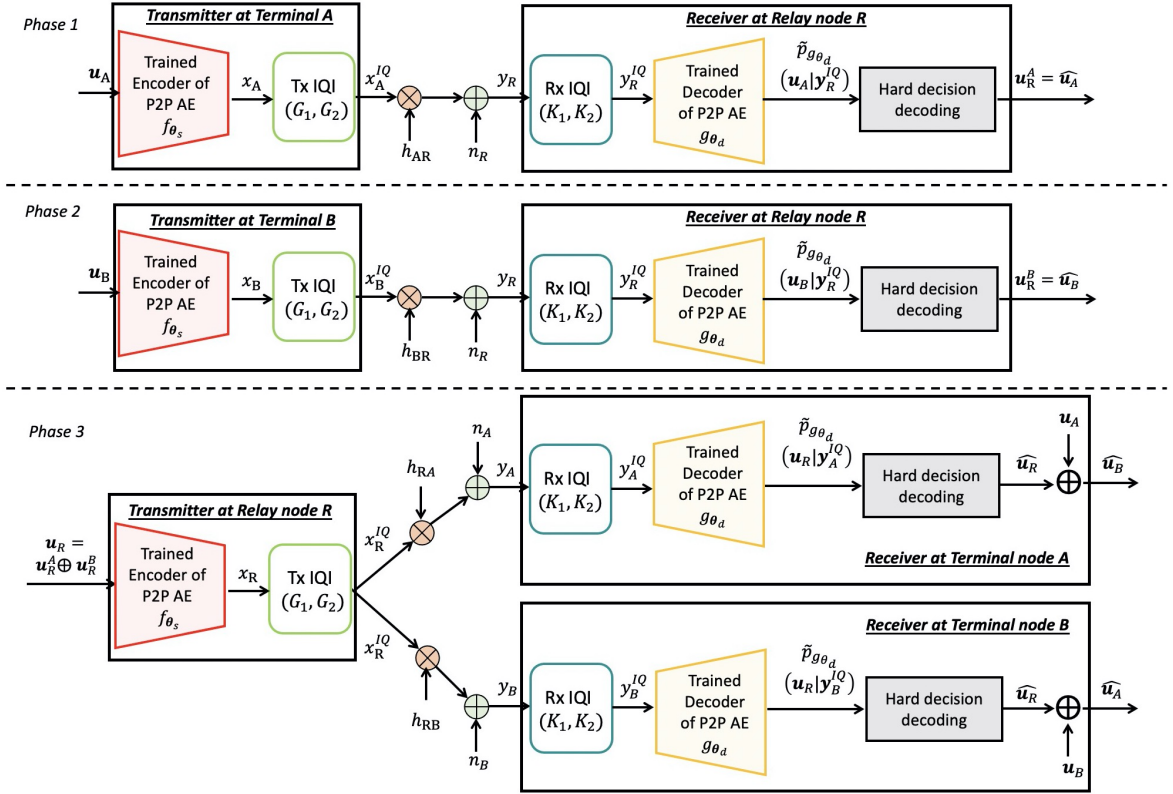


Figure 7.4.2: Testing of proposed stacked bit-wise P2P AE framework for TW-DF relay networks with IQI.

performs block-by-block encoding by mapping \mathbf{u}_A to n complex baseband symbols $\mathbf{x}_A \in \mathbb{C}^n$ using the mapping function of the trained NN encoder of the bit-wise P2P AE $f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s)$, where $\mathbf{u}_s, \mathbf{x}_s = \mathbf{u}_A, \mathbf{x}_A$. Then, the symbol-by-symbol transmission reception takes place between the terminal node A and relay node R. In particular, first the symbol is up-converted in presence of Tx IQI to become x_A^{IQ} using (7.1) with $\Gamma = A$. Then, the signal received by the relay node R becomes y_R using (7.2) with $\Gamma = A$ and $\Upsilon = R$. Lastly, by considering the Rx side IQI, the received signal y_R at the relay node R becomes y_R^{IQ} using (7.4) with $\Upsilon = R$. Once the NN decoder at the relay node R receives n symbols, given by $\mathbf{y}_R^{IQ} \in \mathbb{C}^n$. Then the NN decoder performs block-by-block decoding by de-mapping \mathbf{y}_R^{IQ} to k soft probabilistic outputs $\tilde{p}_{g_{\theta_d}}(\mathbf{u}_A | \mathbf{y}_R^{IQ}) \in [0, 1]^k$, using de-mapping function of trained NN decoder of bit-wise P2P AE at relay node R, given by $g_{\theta_d}(\mathbf{y}_d^{IQ}, \tilde{p}_{g_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d^{IQ}))$, where $\mathbf{u}_s, \mathbf{y}_d^{IQ} = \mathbf{u}_A, \mathbf{y}_R^{IQ}$. Then, soft probabilistic outputs $\tilde{p}_{g_{\theta_d}}(\mathbf{u}_A | \mathbf{y}_R^{IQ}) \in [0, 1]^k$ are converted to k bits given by \mathbf{u}_R^A using hard decision decoding.

- Similar to the first phase, in the second phase, the signal transmission-reception and encoding-decoding using the trained NN encoder and decoder of the bit-wise P2P AE takes place between terminal node B and relay node R, to obtain the k decoded bits \mathbf{u}_R^B at the relay node.

- In the third phase, we propose to employ bit-wise XOR-based encoding of the two decoded signals as

$$\mathbf{u}_R = \mathbf{u}_R^A \oplus \mathbf{u}_R^B \quad (7.20)$$

The relay node R takes k bits $\mathbf{u}_R \in \{0, 1\}^k$ as input and performs block-by-block encoding by mapping \mathbf{u}_R to n complex baseband symbols $\mathbf{x}_R \in \mathbb{C}^n$ using the mapping function of trained NN encoder of bit-wise P2P AE $f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s)$, where $\mathbf{u}_s, \mathbf{x}_s = \mathbf{u}_R, \mathbf{x}_R$. Then relay node broadcasts the signal \mathbf{x}_R symbol-by-symbol, while the signal received at the terminal node A and B in the presence of Tx IQI at relay node and Rx IQI at the terminal nodes A and B can be given by (7.1)–(7.4) with $\Gamma = R, \Upsilon = A$ and $\Gamma = R, \Upsilon = B$, respectively. Once the NN decoder at the terminal nodes $\Gamma = \{A, B\}$ receives n symbols, given by $\mathbf{y}_\Gamma^{IQ} \in \mathbb{C}^n$. Then the NN decoder at each of the terminal node Γ performs block-by-block decoding by de-mapping \mathbf{y}_Γ^{IQ} to k soft probabilistic outputs $\tilde{p}_{g_{\theta_d}}(\mathbf{u}_R | \mathbf{y}_\Gamma^{IQ}) \in [0, 1]^k$, by using the de-mapping function of the trained NN decoder of bit-wise P2P AE at each terminal node Γ , given by $g_{\theta_d}(\mathbf{y}_d^{IQ}, \tilde{p}_{g_{\theta_d}}(\mathbf{u}_s | \mathbf{y}_d^{IQ}))$, where $\mathbf{u}_s, \mathbf{y}_d^{IQ} = \mathbf{u}_R, \mathbf{y}_\Gamma^{IQ}$. Then, at each terminal node Γ the soft probabilistic outputs $\tilde{p}_{g_{\theta_d}}(\mathbf{u}_R | \mathbf{y}_\Gamma^{IQ}) \in [0, 1]^k$ are converted to k bits, given by $\hat{\mathbf{u}}_R$, using hard decision decoding. Since each terminal node has its information bits, it can now perform bit-wise XOR on the decoded bits and its bits to decode the other terminal node's bits, as follows:

$$\text{At terminal node A :} \quad \hat{\mathbf{u}}_B = \hat{\mathbf{u}}_R \oplus \mathbf{u}_A \quad (7.21)$$

$$\text{At terminal node B :} \quad \hat{\mathbf{u}}_A = \hat{\mathbf{u}}_R \oplus \mathbf{u}_B \quad (7.22)$$

7.5 Performance Evaluation

In this section, we evaluate the proposed stacked AE frameworks for the OW-DF and TW-DF relay networks with IQI. We utilize Rayleigh block fading channels with $(n, k) = (7, 8)$, i.e., rate $R = 8/14$ (for OW-DF) and rate $R = 16/21$ (for TW-DF), where the channel remains constant for $n = 7$ symbols and then changes randomly. For the conventional scenarios, we utilize QPSK (in the presence of CSI) and d-QPSK (in the absence of CSI) with $(7, 4)$ Hamming codes and consider the following as benchmarks:

- *MLD: No IQIC* – Herein, we consider the absence of IQI parameters information and use the MLD.
- *MLD: ZF IQIC* – Herein, we consider perfect knowledge of IQI parameters; thus, we perform ZF-based IQI compensation and use the MLD, as detailed in Sec. 7.2.

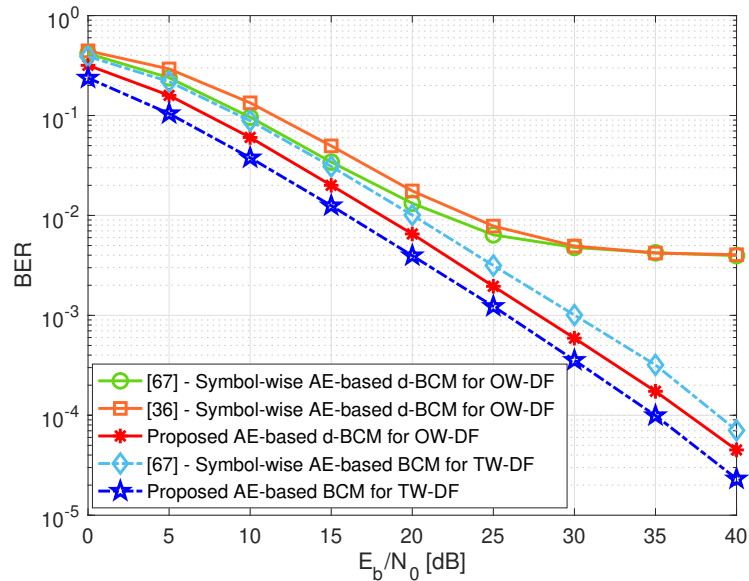


Figure 7.5.1: Comparison of proposed stacked AE frameworks with state-of-the-art AE frameworks in [36], [67] in an ideal scenario.

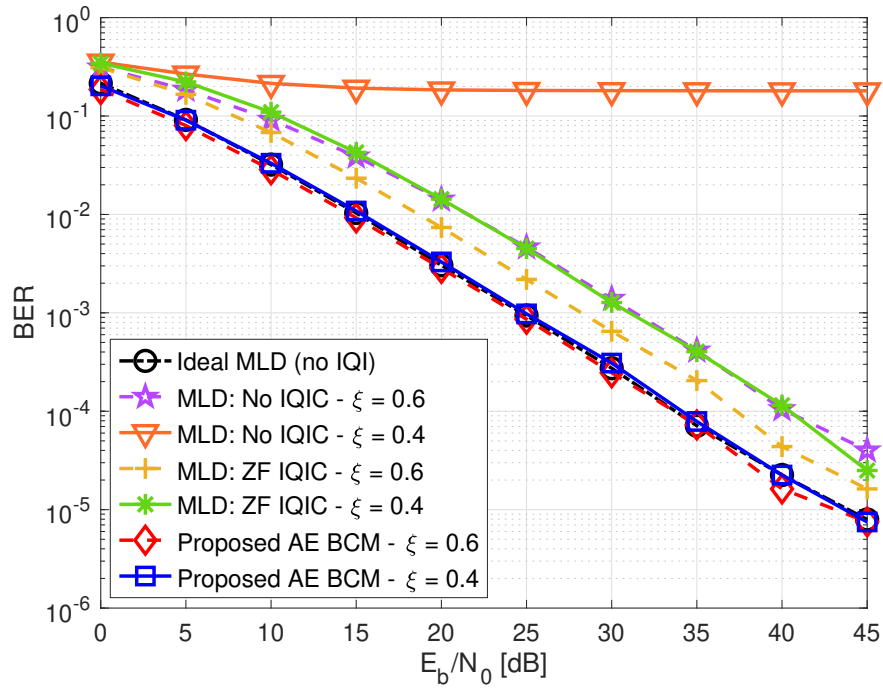
- *Ideal MLD (no IQI)* – Herein, we consider an ideal relay network with no IQI, and use the MLD.

7.5.1 Comparison of Proposed Stacked AE-based frameworks with State-of-The-Art AE works

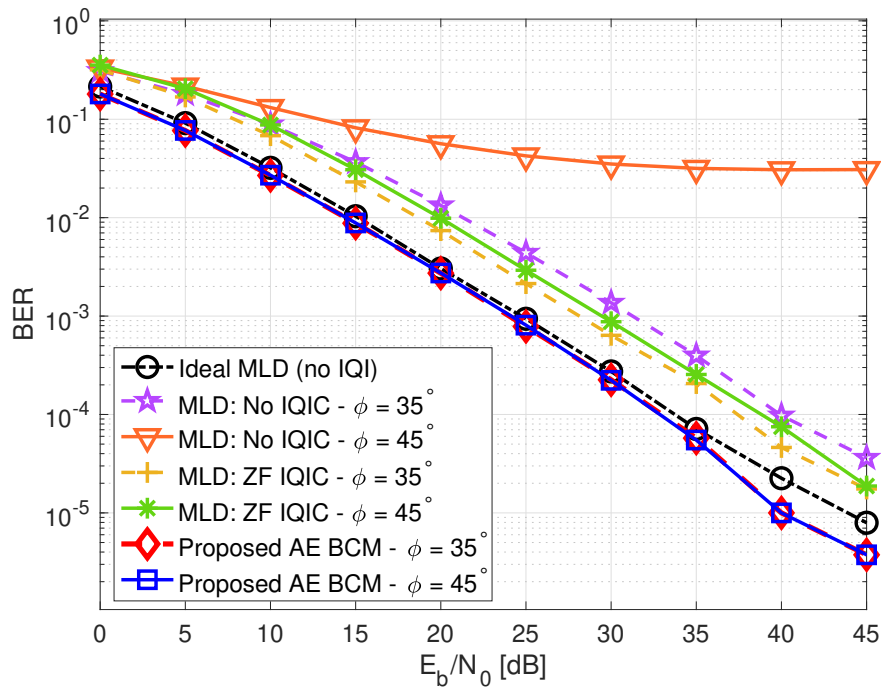
In Fig. 7.5.1, we compare the proposed stacked AE frameworks with state-of-the-art AE works in [36], [67] for OW-DF and in [67] for TW-DF relay networks. Moreover, we consider ideal OW-DF and TW-DF relay networks without IQI. This will remove all the additional gains achieved by the proposed AE frameworks designed to mitigate the IQI.

The authors in [36] and [67] proposed a symbol-wise AE-based d-BCM for the OW-DF relay networks using an iterative two-step training policy and two-step training policy, respectively. In Fig. 7.5.1, we compare the proposed stacked bit-wise AE and bit-wise denoising AE framework-based d-BCM with the d-BCM proposed in [36], [67]. Clearly, the proposed stacked bit-wise AE framework is able to decode the signal with increasing E_b/N_0 and outperform [36], [67], while the performance of [36], [67] reaches error floor around $10^{-2.5}$.

The authors in [67] proposed a symbol-wise AE-based BCM for TW-DF relay networks using an iterative three-step training policy for separate symbol-wise AEs in each transmission phase. In Fig. 7.5.1, we compare the proposed stacked bit-wise P2P AE framework-based BCM with the BCM proposed in [67] for the TW-DF relay networks. Clearly, the proposed stacked bit-wise P2P AE outperforms the [67] by 5 dB, without



(a) Analyzing BCM designs for different amplitude offsets and fixed phase offset $\phi = 25^\circ$.



(b) Analyzing BCM designs for different phase offsets and fixed amplitude offset $\xi = 0.7$.

Figure 7.5.2: Stacked bit-wise AE and denoising bit-wise AE-based BCM designs for OW-DF relay networks with IQI.

using separate AE training in each phase by only re-utilizing the bit-wise P2P AE frameworks, also reducing the computational complexity and training time-cost.

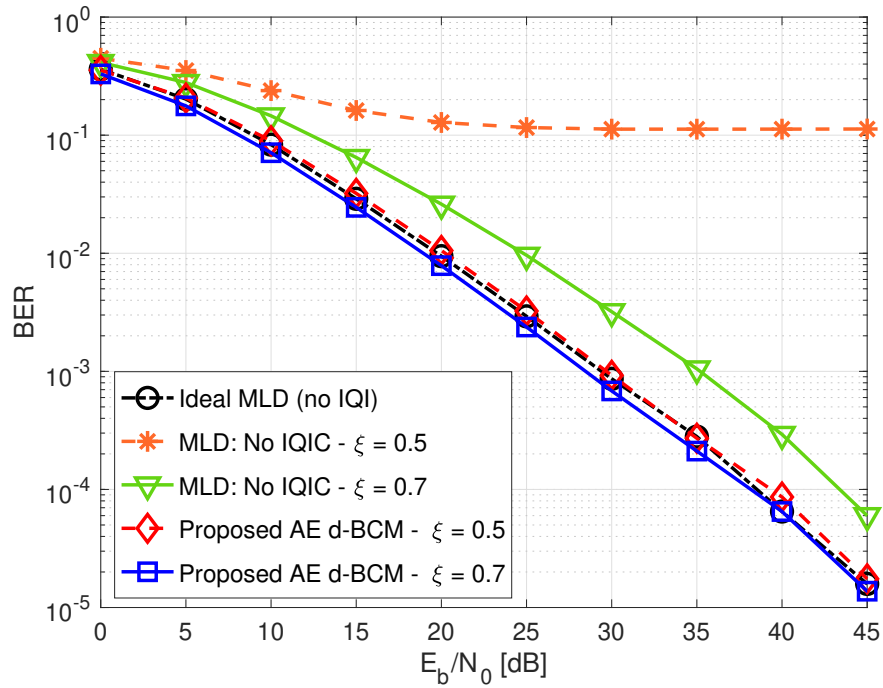
In general, these gains are achieved because of the proposed stacked bit-wise AE frameworks performing the automatic bit-labeling and maximization of the bit-wise mutual information (MI). In contrast, the symbol-wise AE performs heuristic bit-labeling by solving the $2^k!$ combinatorial problem and maximization of the symbol-wise MI. Thereby showing the merits of proposed stacked bit-wise AE frameworks for OW-DF and TW-DF relay networks.

7.5.2 BER Evaluation for the BCM and d-BCM Designs for the OW-DF Relay Networks with IQI

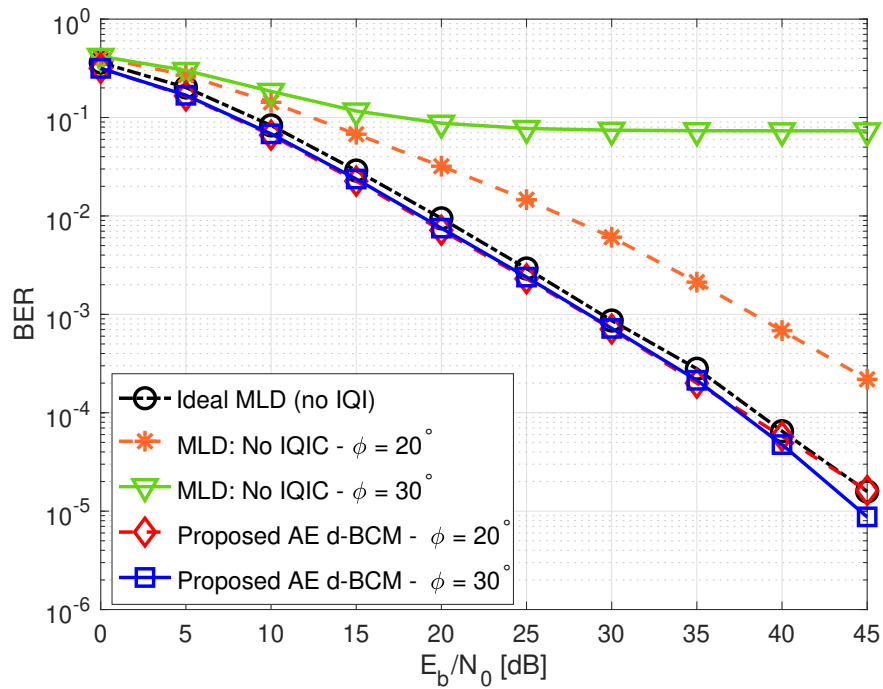
In Fig. 7.5.2a, 7.5.2b, we train a single proposed stacked bit-wise AE and bit-wise denoising AE-based BCM designs with CSI knowledge for the OW-DF relay networks, and analyze by varying the amplitude offsets (in Fig. 7.5.2a) and phase offsets (in Fig. 7.5.2b). The MLD with ZF-based IQI compensation is always able to decode the signals because of the presence of IQI parameters information. Moreover, when the amplitude offset is low $\xi = 0.6$ (in Fig. 7.5.2a) or phase offset is low $\phi = 35^\circ$ (in Fig. 7.5.2b) the MLD without any IQI compensation (MLD: No IQIC) is able to decode the signals because SIR still remains 4.5 dB and 3.9 dB, respectively. However, when the amplitude offset is high $\xi = 0.4$ (in Fig. 7.5.2a) or phase offset is high $\phi = 45^\circ$ (in Fig. 7.5.2b) the MLD without any IQI compensation (MLD: No IQIC) is unable to decode the signals because SIR becomes 2.2 dB and 2.6 dB, respectively. Thus, when the SIR becomes lower than 3 dB MLD is unable to decode the signals because IQI becomes too high.

In Fig. 7.5.2a, 7.5.2b, we can see that the proposed stacked AE-based BCM design is always able to decode the signal even for $\text{SIR} \leq 3$ dB, even without utilizing the IQI parameters information. The proposed stacked AE-based BCM design performs similar to MLD for an ideal OW-DF relay network without IQI (Ideal MLD), indicating that the proposed BCM designs almost completely remove the IQI without utilizing the IQI parameters information, even under low SIR regimes.

In Fig. 7.5.3a, 7.5.3b, we train a single proposed stacked bit-wise AE and bit-wise denoising AE-based d-BCM designs without the CSI knowledge for the OW-DF relay networks, and analyze by varying the amplitude offsets (in Fig. 7.5.3a) and phase offsets (in Fig. 7.5.3b). Similar to BCM design when the amplitude offset or phase offset is low $\xi = 0.7$ and $\phi = 20^\circ$ in Fig. 7.5.3a and Fig. 7.5.3b, respectively, the MLD is able to decode the signals because SIR is still high 12.9 dB and 11.5 dB, respectively. However, when the amplitude offset or phase offset is high $\xi = 0.5$ and $\phi = 30^\circ$ in Fig. 7.5.3a and Fig. 7.5.3b, respectively, the MLD is unable to decode the signals because SIR be-



(a) Analyzing d-BCM designs for different amplitude offsets and fixed phase offset $\phi = 10^\circ$.



(b) Analyzing d-BCM designs for different phase offsets and fixed amplitude offset $\xi = 0.8$.

Figure 7.5.3: Stacked bit-wise AE and denoising bit-wise AE-based d-BCM designs for OW-DF relay networks with IQI.

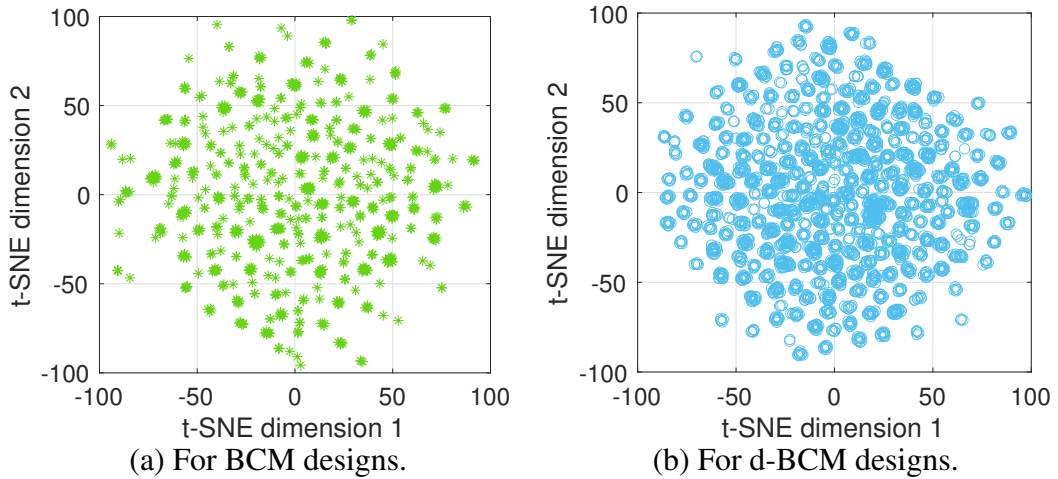


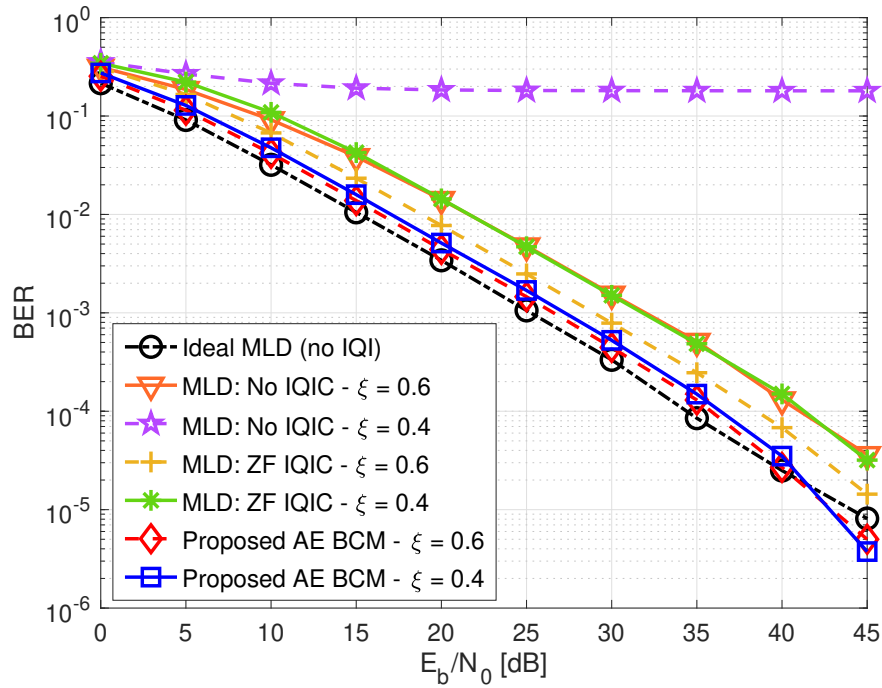
Figure 7.5.4: t-SNE plots for NN encoder at the relay node in bit-wise denoising AE framework of OW-DF relay network.

comes low as 4.3 dB and 5.9 dB, respectively. Thus, when the SIR becomes lower than 6 dB MLD is unable to decode the signals because IQI becomes too high, and the IQI parameters information and CSI knowledge is absent.

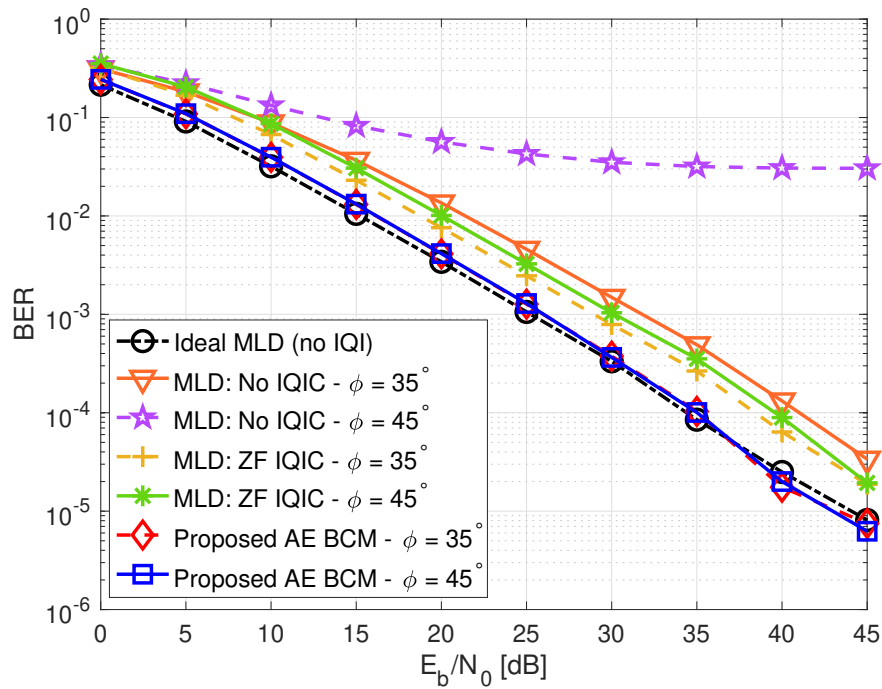
In Fig. 7.5.3a, 7.5.3b, we can see that the proposed stacked AE-based d-BCM design is always able to decode the signal even for $\text{SIR} \leq 6$ dB, without utilizing IQI parameter information and CSI knowledge. The proposed stacked AE-based d-BCM designs perform similar to the MLD for an ideal OW-DF relay network without IQI (Ideal MLD), indicating that the proposed stacked bit-wise AE-based d-BCM almost completely removes the IQI without utilizing the IQI parameters information and CSI knowledge, even under low SIR regimes.

Please note in Fig. 7.5.2, 7.5.3, the proposed AE-based BCM and d-BCM designs almost completely remove the IQI, without IQI parameters information, this is because of the following reasons:

- In a conventional scenario, we employ QPSK and d-QPSK modulations with a minimum Euclidean distance between symbols as 1.4 and 0.76, respectively. Also, we employ the $(7, 4)$ Hamming code since $k = 8$ thus it provides up to 2 bits of error correction capabilities.
- Now, we focus on the bit-wise AE employed in the first phase transmission of the OW-DF relay networks. We see that the bit-wise AE forms 2^k codewords for each of the possible codewords because we formulate the optimization problem as a multi-label binary classification problem with k labels each having a possible 0/1 binary value. These 2^k codewords are designed in the $2n$ -dimensional space because the NN encoder has the last dense layer that outputs $2n$ real values. Also, we note that these 2^k designed codewords have the normalized fourth-order moment as

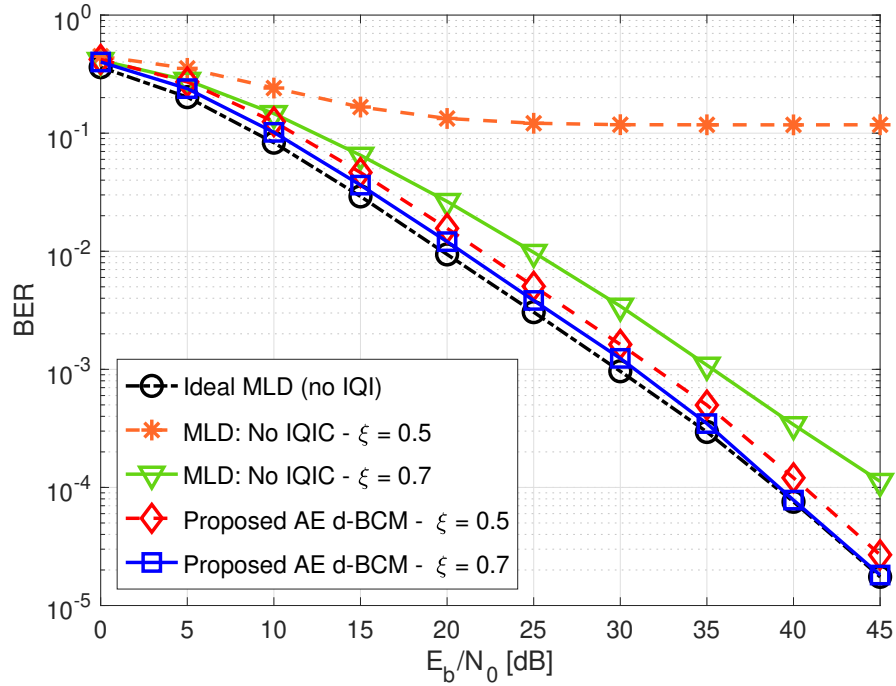


(a) Analyzing BCM designs for different amplitude offsets and fixed phase offset $\phi = 25^\circ$.

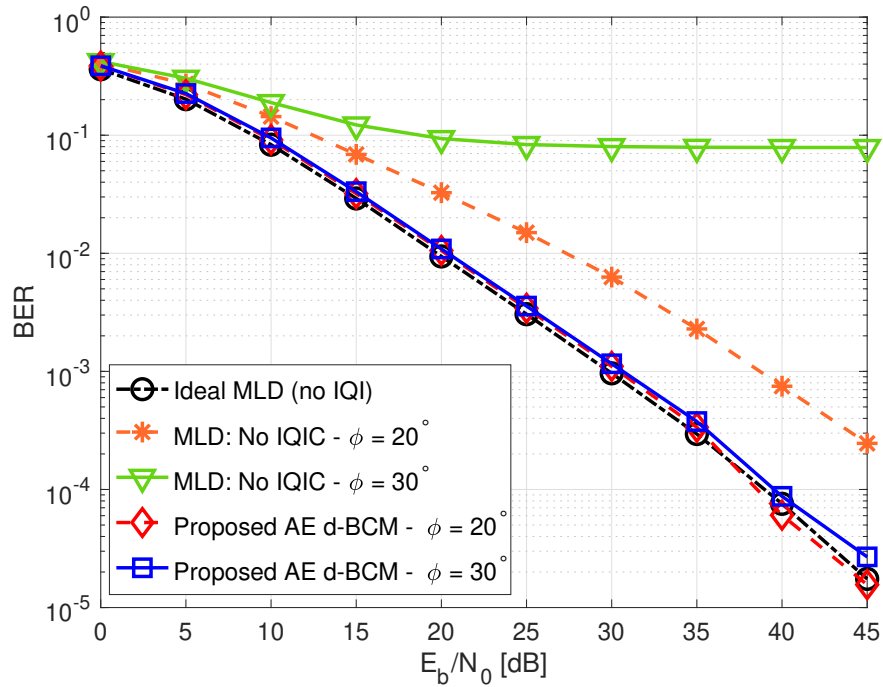


(b) Analyzing BCM designs for different phase offsets and fixed amplitude offset $\xi = 0.7$.

Figure 7.5.5: Stacked bit-wise P2P AE-based BCM designs for the TW-DF relay networks with IQI.



(a) Analyzing d-BCM designs for different amplitude offsets and fixed phase offset $\phi = 10^\circ$.



(b) Analyzing d-BCM designs for different phase offsets and fixed amplitude offset $\xi = 0.8$.

Figure 7.5.6: Stacked bit-wise P2P AE-based d-BCM designs for the TW-DF relay networks with IQI.

1, indicating that Spherical codes are formed. We also know that spherical codes are optimal for a small block length (n) [74]. From the modulation perspective, bit-wise AE leads to the maximization of the minimum Euclidean distance between codewords to 1.5 and 1.2 for BCM and d-BCM designs, respectively. From the coding perspective, bit-wise AE leads to the maximization of the minimum Hamming distance between the codewords in the $2n$ -dimensional space.

- Now, we focus on the bit-wise denoising AE employed in the second phase transmission of the OW-DF relay networks. Since the bit-wise denoising AE input is soft probabilistic outputs of the first hop's bit-wise AE, the NN encoder of the bit-wise denoising AE does not form 2^k codewords, instead almost a slightly different codeword is learned for different soft probabilistic outputs. This possibly helps the bit-wise denoising AE remove the noise in the input soft probabilistic outputs while decoding the signal at Terminal B's NN decoder. In Fig. 7.5.4, we show the t-stochastic neighbor embedding (t-SNE) diagrams (as detailed in *Definition 15*) to visualize the codewords designed by the bit-wise denoising AE-based BCM and d-BCM designs in $2n$ -dimensional space.

Thus, by intelligent BCM or d-BCM designs, a single proposed stacked bit-wise AE and bit-wise denoising AE framework can remove all the varying levels of IQI in an OW-DF relay network almost wholly, without IQI parameters information and/or CSI knowledge.

7.5.3 BER Evaluation for the BCM and d-BCM Designs for the TW-DF Relay Networks with IQI

In Fig. 7.5.5a, 7.5.5b, we train a single proposed stacked bit-wise P2P AE-based BCM design with CSI knowledge for the TW-DF relay networks, and analyze by varying amplitude offsets (in Fig. 7.5.5a) and phase offsets (in Fig. 7.5.5b).

Also, in Fig. 7.5.6a, 7.5.6b, we train a single proposed bit-wise P2P AE-based d-BCM designs without the CSI knowledge for the TW-DF relay networks, and analyze by varying amplitude offsets (in Fig. 7.5.6a) and phase offsets (in Fig. 7.5.6b).

In Fig. 7.5.5,7.5.6, we see exactly similar trends for the performance improvements as the BCM and d-BCM designs for the OW-DF relay networks in Fig. 7.5.2, 7.5.3. Please note since IQI settings remain same as Fig. 7.5.2, 7.5.3, thus SIR levels also remain the same. In summary, the MLD is unable to decode the signals when $\text{SIR} \leq 3$ dB (in the presence of CSI) and $\text{SIR} \leq 6$ dB (in the absence of CSI), while the proposed stacked bit-wise P2P AE-based BCM and d-BCM can perform similar to ideal TW-DF relay networks, indicating that the proposed stacked bit-wise P2P AE can completely remove the deteriorating impacts of IQI, without utilizing the IQI parameters information and/or CSI knowledge, even in extremely low SIR regimes.

Please note that since we train a single bit-wise P2P AE and employ it in a stacked form to make a TW-DF relay network, the explanation of performance gains remains the same as the bit-wise AE in the first phase transmission of the OW-DF relay networks detailed in Sec. 7.5.2. Furthermore, please note since we employ the bit-wise P2P AE in stacked form, this explanation remains the same for the NN encoders at all the nodes (A, R, B) in the TW-DF relay networks. Thus, by intelligent BCM or d-BCM designs, a single proposed stacked bit-wise P2P AE framework can remove all the varying levels of IQI in a TW-DF relay network almost wholly, without IQI parameters information and/or CSI knowledge.

7.6 Conclusion

In this chapter, firstly, we propose a stacked AE-based BCM and d-BCM designs for the OW-DF relay network in the presence of IQI at all the nodes. We propose to employ bit-wise AE in the first transmission phase and introduce a bit-wise denoising AE for second phase transmission. We also propose a new training policy for bit-wise denoising AE. Secondly, we propose a stacked AE-based BCM and d-BCM designs for the TW-DF relay network in the presence of IQI at all the nodes. We propose to employ a bit-wise P2P AE framework in a stacked form to make a TW-DF relay network and utilize the benefits of conventional bit-wise XOR-based encoding-decoding. For both the stacked AE-based OW-DF and TW-DF relay networks, we propose a single trained AE framework-based BCM and d-BCM designs that can generalize well on any levels of testing IQI and signal-to-noise-ratio (SNR). We remove the need for IQI parameter information to decode the signals even in extremely low SIR regimes and high transmission rates for both the stacked AE-based OW-DF and TW-DF relay networks. Thereby reducing the feedback overhead while achieving remarkable BER performance gains.

Chapter 8

AE-based Multi-User Downlink Networks

8.1 Introduction

Designing an appropriate multiple access technique is one of the requisite notions of improving the system capacity. We can broadly categorize the multiple access schemes as – orthogonal multiple access (OMA) and non-orthogonal multiple access (NOMA), as shown in Fig. 8.1.1. In the OMA scheme, different signals from different users remain orthogonal to each other, enabling a perfect receiver to remove all the unwanted signals. Its typical example includes time-division multiple access and orthogonal frequency division multiple access. On the other hand, in the NOMA scheme, various users are served in orthogonal resource blocks by multiplexing the signals in the power, code, or frequency domain. In other words, NOMA enables numerous users to be served on one frequency channel simultaneously within the same cell. In contrast to OMA, NOMA offers significant advantages – improved spectral efficiency (SE), higher cell-edge throughput, and low latency communication [105]. Broadly, NOMA can be classified into two major categories, power-domain NOMA (PD-NOMA) and code-domain NOMA (CD-NOMA). In this chapter, consider the PD-NOMA as the benchmark, for readability hereby referred to as conventional NOMA, wherein the signal of the strong user is allocated lesser power than the signal of the weak user and superposed before transmission. Later, a weak user decodes its signal directly. In contrast, the strong user has to perform successive interference cancellation (SIC) to decode and remove the weaker user’s signal before decoding its signal [105], [106].

Remark 29 *Please note that although the abbreviation SIC represented the self-interference cancellation in previous Chapters 5, 7. In this chapter, the abbreviation SIC represents the “successive interference cancellation”.*

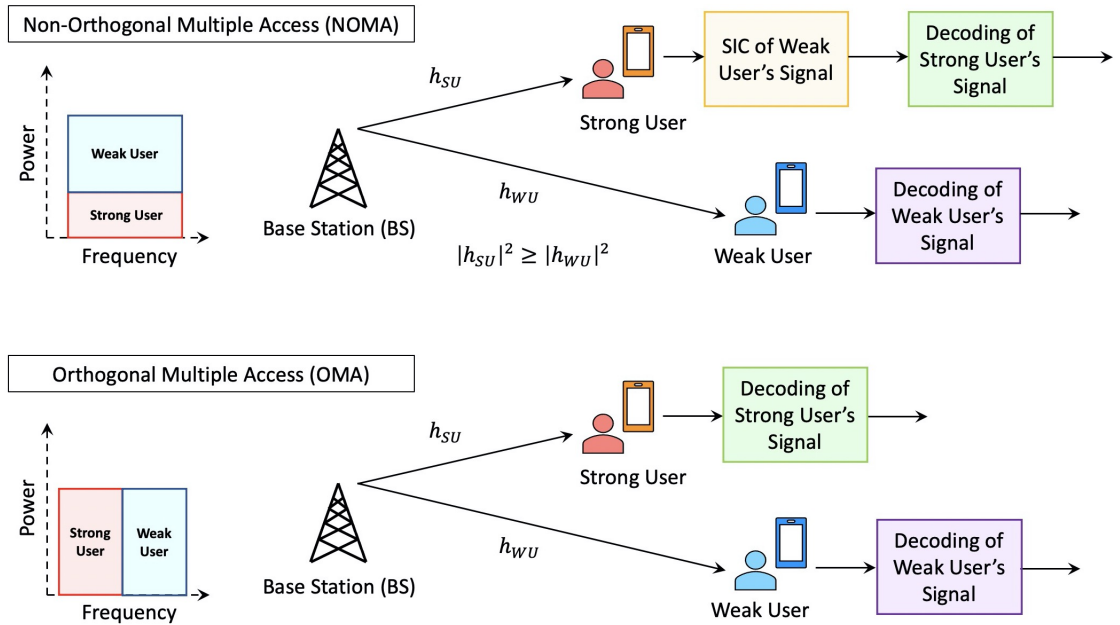


Figure 8.1.1: An illustration of signal transmission–reception in non-orthogonal multiple access and orthogonal multiple access networks.

Despite the proven benefits of NOMA, there are certain practical challenges for its deployment [106], as detailed below

- The first challenge is its high sensitivity to the users' channel state information (CSI) knowledge. The base station (BS) receives the CSI gains of the users and then decides the power allocation factor (PAF) for each user, which should be optimal for achieving the SE gains of NOMA.
- Secondly, the paired users need to be informed of the SIC order and the PAF; this leads to the lapse of some time slots, resulting in data rate degradation.
- Thirdly, the minimum SNR gap between the strong and weak users should be approximately 8 dB (at least 3 dB) for satisfactory implementation of SIC at the stronger users [107].
- Lastly, SIC requires additional implementation complexity. Also, intuitively, if multiple users are served in a cluster using NOMA, then the stronger users will implement multiple SICs serially removing each of the weaker users' signals, thereby increasing the chances of error propagation because of imperfect SICs.

Therefore, in this chapter, we aim to utilize the end-to-end learning-based data-driven autoencoder (AE) frameworks for tackling these limitations of the PD-NOMA network in a multi-user downlink network.

The optimal constellation designs by optimizing the PAF have been widely investigated for realizing the full potential of NOMA networks. However, the PAF is deter-

mined using the signal-to-noise ratio (SNR) of all the downlink users. Recently, an optimal inter-constellation rotation was proposed using the minimum distance criterion for an uplink PD-NOMA network that enabled SNR independent system formulation [108]. Moreover, for the \mathcal{C}_D -NOMA networks, the block coded modulation (BCM) design by optimizing the codewords with multi-user detection has shown promising performance gains [109]. Furthermore, optimal coded constellation design for two-user PD-NOMA is analyzed in [110] by utilizing the gradient relationship between the mutual information (MI) and minimum mean squared error. Constellation rotation-based design for PD-NOMA in simultaneous wireless information and power transfer systems has also been shown to efficiently increase the harvested energy without incurring additional symbol detection errors [111]. In summary, the optimal constellation rotation and BCM designs have appeared promising solutions for realizing the full potential of the NOMA networks.

In previous Chapters 2–7, we have shown the potential benefits of employing an AE-based framework for performing BCM and differential BCM (d-BCM) designs for rate $R = k/n$ [bits/channel-reuse], leading to significant bit-error-rate (BER) performance gains. Recently, AE frameworks have also been shown to improve the performance of the CD-NOMA networks [112, 113, 114, 115] and PD-NOMA networks [116, 117, 118, 39]. The AE-based PD-NOMA network is proposed using a bit-wise AE [116, 39] and symbol-wise AE [118, 117] frameworks. Within the AE framework, the input bits at the BS have been superposed using a single neural network (NN)-based encoder [116, 117, 39], and using multiple NN-based encoders (one for each user) [118].

Further, the recent works [116, 117, 118, 39] focussed on the AE-based 2-dimensional modulation designs for a two-user downlink scenario for the PD-NOMA networks. However, the complexity in optimal constellation design increases with the number of users [108, 109, 110, 111, 119]. Therefore it becomes critical, especially for IoT applications, to develop frameworks that can design optimal constellations for a more significant number of users. Moreover, similar to conventional PD-NOMA, recent AE frameworks assume an SNR gap of 3 dB between the multiplexed users [116, 118, 39] (which remains a similar to the implementation of SIC in conventional networks). At the same time, [117] focuses on broadcast channels. Therefore, it is crucial to develop an AE framework that remains unaffected by the SNR gaps between the users, improving the BER performance even when all the users have the same receiving SNR.

While [117, 116, 118] focussed on the AE framework for the additive white Gaussian noise (AWGN) channel, and the fading channels are investigated in [39]. All these works require channel (signal) powers (of each user) at the BS to optimize and use the PAF for signal superposition, respectively. Thus the feedback overhead in their designed network remains the same as conventional PD-NOMA. Additionally, the optimization for determining the PAF needs to be performed separately at the BS.

In this chapter, we propose a bit-wise Fixed AE-based framework for multi-user downlink networks. We broadly classify the proposed AE framework as follows:

Definition 23 (Single encoder multiple decoders (SEMD)) *Within the SEMD-based AE framework, we employ a single NN-based encoder at the base station that takes all the downlink user's information bits together as input and generates superposed complex baseband symbol(s) as output. While each downlink user has an individual NN-based decoder, decoding its information bits.*

Definition 24 (Multiple encoders multiple decoders (MEMD)) *Within the MEMD-based AE framework, at the base station, we employ a separate NN-based encoder for encoding each user's information bits to complex baseband symbol(s), which are superposed simply by addition before transmission. While each downlink user has an individual NN-based decoder, decoding its information bits.*

The major contributions of this chapter, in contrast to the existing literature (as detailed in Table 8.1.1), are summarized below

- We propose a bit-wise AE-based SEMD and MEMD frameworks for downlink multi-user networks. With SEMD and MEMD frameworks, we remove the utilization of PAF at BS and the multiple SIC at strong users while operating under fading channels. Thereby removing the need to feedback channel gains to the BS, PAF optimization, error propagation due to multiple SIC, and the need to send the SIC ordering information to the users.
- We impose the equal QoS constraint for each user, removing the concept of strong and weak users. Thus, our proposed AE frameworks remain unaffected by the SNR gaps between the users and we can decode the superposed signal of multiple users on the same SNR. We present the analysis when the SNR gap between the users is $\{0, 6\}$ dB. For practical purposes, we consider Rayleigh fading channels. We also show that the performance gains of the proposed AE frameworks increase with the number of users.
- We design a generalizable NN-based decoder for the multi-user scenario, such that it remains the same for all the users, even if the number of users increases, but also for both the AE-based SEMD and MEMD frameworks.
- For both the SEMD and MEMD frameworks, we propose an AE-based 2-dimensional modulation design under Rayleigh block fading (RBF) channels for two, three, and four users' scenarios. For greater insights, we analyze the constellation design and bit-labeling obtained at the NN encoder of the BS. Specifically, we compare the

Table 8.1.1: Comparison of our chapter with state-of-the-art works [116, 117, 118, 39].

<i>Ref</i>	<i>Encoders</i>	<i>Channel</i>	<i>PAF re-quired</i>	<i>Decoding decision</i>	<i>DL-mod-ulation</i>	<i>AE-BCM AE-d-BCM</i>	<i>CSI required</i>	<i>No. of users</i>	<i>SNR gap</i>	<i>bit ma-pping</i>
[116]	Single	AWGN	✓	Soft	✓	✗	N/A	2	≥ 3	✗
[118]	Multiple	AWGN	✓	Hard	✓	✗	N/A	2	≥ 3	✗
[117]	Single	AWGN	✗	Hard	✓	✗	N/A	2	0	✗
[39]	Single	Rayleigh	✓	Soft	✓	✗	✓	2	≥ 3	✗
<i>Ours</i>	Single; Multiple	Rayleigh	✗	Soft	✓	✓	No for d-BCM	2, 3,4	0; 6	✓

minimum squared Euclidean distances establishing the reasons for user fairness and performance gains.

- For both the AE-based SEMD and MEMD frameworks, we propose an AE-based BCM design with the CSI knowledge. In contrast to conventional NOMA, where CSI is ‘mandatory’ to decode the multiplexed signals at the users, we altogether remove the CSI requirement by proposing AE-based d-BCM designs for both the SEMD and MEMD frameworks. For greater insights in the BCM and d-BCM designs, we utilize the t-stochastic neighbor embedding (t-SNE) to convert the code-words in $2n$ -dimensional space to 2-dimensional space and utilize k-medoids clustering to map the symbols and analyze the encoding at the BS.
- Lastly, we analyze higher modulation scenarios, i.e., QPSK+QPSK, QPSK+BPSK, and 16QAM+BPSK, and evince the performance gains of the proposed AE-based frameworks.

8.2 Conventional Power-Domain Downlink NOMA

We consider a J -user downlink NOMA network as shown in Fig. 8.2.1, wherein the BS wants to transmit the superposed signal to the J users using the PD-NOMA. Thus, the BS performs superposition of the signals of the J downlink users. In particular, the information bits $\mathbf{u}_j = \{0, 1\}^k$ of each j^{th} downlink user, where $j = \{1, \dots, J\}$, are mapped to a complex baseband symbol $x_j = g_m(\mathbf{u}_j) \in \mathbb{C}$, where $g_m(\cdot)$ denotes the modulation process.

Let us denote the RBF channel between the BS and each j^{th} downlink user by $h_j \in \mathcal{CN}(0, 1)$ and $n_j \in \mathcal{CN}(0, \sigma_j^2)$ represent the AWGN at each j^{th} downlink user. Let P denote the total transmission power of the BS and α_j^P represent the power allocation factor for the j^{th} downlink user, such that $\sum_{j=1}^J \alpha_j^P = 1$. For the sake of simplicity, we assume that the receiving SNR of the J downlink users are in a descending order i.e.

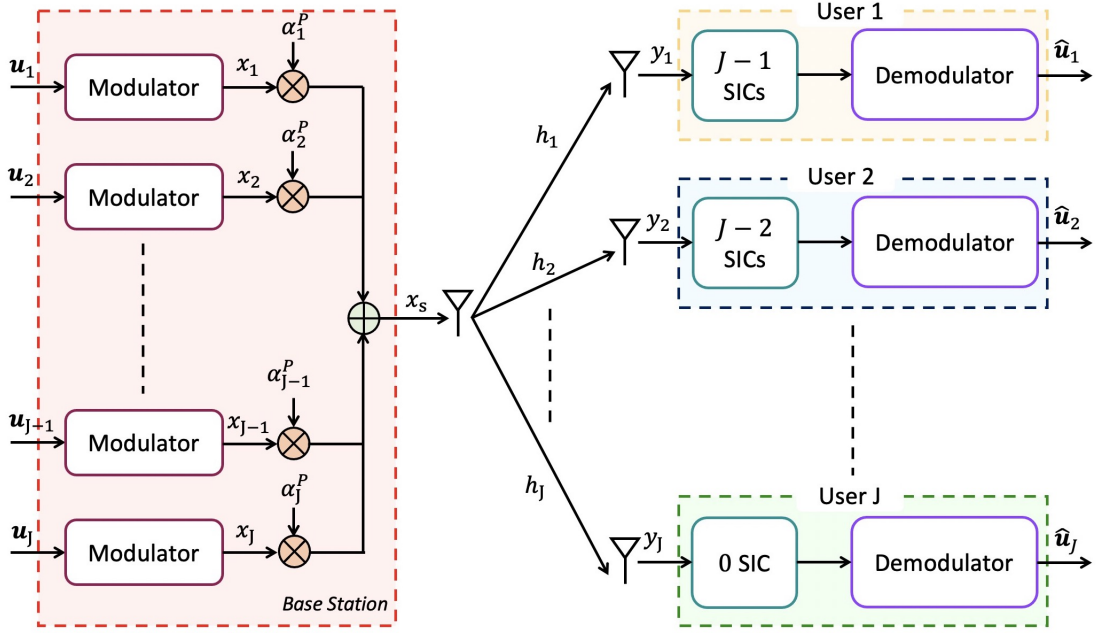


Figure 8.2.1: Conventional power-domain downlink NOMA for multi-user scenario

$\frac{|h_1|^2}{\sigma_1^2} > \dots > \frac{|h_J|^2}{\sigma_J^2}$, thus the superposed signal is given by

$$x_s = \sum_{j=1}^J \sqrt{P\alpha_j^P} x_j \quad (8.1)$$

where the PAF for the j^{th} user remains $\alpha_1^P < \alpha_2^P < \dots < \alpha_J^P$. Moreover, the superposed symbol is normalized such that $E[|x_s|^2] = 1$. The superposed symbol is broadcasted to all the J users, and the received signal at the j^{th} downlink user is given by

$$\begin{aligned} y_j &= h_j x_s + n_j, \quad \forall j \\ &= \underbrace{\sqrt{P\alpha_j^P} h_j x_j}_{\text{Own signal}} + \underbrace{\sum_{i=1, i \neq j}^J \sqrt{P\alpha_i^P} h_j x_i}_{\text{Other users signal}} + \underbrace{n_j}_{\text{Noise}} \end{aligned} \quad (8.2)$$

Without loss of generality, the J^{th} downlink user is assigned the maximum power by maximizing α_J^P , such that it can decode its signal treating all the other $J - 1$ user's signals as noise, whereas all the other users first implement SIC to decode and subtract the signals of the users with higher receiving SNR, and then they decode their signals. Thus, for instance, user 2 will decode and subtract the signals from user 3 to user J (by implementing $J - 2$ SICs in total) and then decode its signal. Therefore, the total number of SICs that need to be performed increases exponentially with the number of multiplexed users in a downlink PD-NOMA network.

8.3 Proposed AE-based Multi-user Downlink Network

In this section, we propose an AE-based multi-user downlink network. Specifically, we propose NN-based encoder(s) at the BS that superposes the k_j , for all $j = \{1, \dots, J\}$, bits of J users to form n complex baseband symbols. Then BS performs symbol-by-symbol broadcasting to all the J downlink users (simultaneously). While each user is proposed to have an individual NN-based decoder that helps in decoding its respective k_j bits. Thus, the rate, as detailed in Definition 1, of the J user downlink network becomes $R = \sum_{j=1}^J k_j/n$ [bits/channel-reuse].

In this chapter, we utilize L fully-connected (dense) layers in the NN encoder-decoder pair of the AE framework, wherein the output of the $l^{\text{th}} \in \{1, \dots, L\}$ dense layer $\omega_l \in \mathbb{R}^{\delta_l}$ can be given as (2.2). We also utilize L' batch-normalization layers \mathcal{B}_N [120] in the NN decoder(s) of the AE framework. In the $l'^{\text{th}} \in \{1, \dots, L'\}$ \mathcal{B}_N layer, mean (μ_B) and variance (σ_B^2) of the samples in input batch are used to normalise the input tensor with the calculated statistics, which is then scaled and shifted, where the output of \mathcal{B}_N layer is given as follows:

$$\mathcal{B}_N(\omega_{l'}) := \omega_{l'} = \varphi \left(\omega_{l'} - \frac{\mu_B}{\sqrt{\sigma_B^2 - \epsilon}} \right) + \varrho \quad (8.3)$$

where the notation $:=$ is ‘represents’, and φ and ϱ are hyper-parameters learned during the training phase. Please note that the dense layers help us in extracting the non-linear relationships with the help of activation functions, while the \mathcal{B}_N layer helps us in decreasing the covariance shift of the hidden nodes, thereby maintaining the statistical variance of the training phase in the testing phase as well. Now, we elaborate on designing the proposed AE-based SEMD and MEMD frameworks in detail below.

8.3.1 Single Encoder Multiple Decoders (SEMD)

In this subsection, we detail the AE-based SEMD framework for multi-user downlink network, as shown in Fig. 8.3.1.

Designing of a Single NN-based Encoder at the BS

The base station comprises of a NN-based encoder that takes the concatenated bits of all the J users as input, given by $\mathbf{u}_s = [\mathbf{u}_1, \dots, \mathbf{u}_J] \in \{0, 1\}^{\sum_{j=1}^J k_j}$, where $\mathbf{u}_j \in \{0, 1\}^{k_j}$, for each j^{th} downlink user, and maps it to n complex baseband symbols $\mathbf{x}_s \in \mathbb{C}^n$. Thereby, performing joint block channel-coding, modulation design, and signal superposition. In particular, the encoder designs a NN-based mapping function $f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s) : \mathbf{u}_s \rightarrow \mathbf{x}_s$ where θ_s represents the optimization parameters including weights and bias terms of the

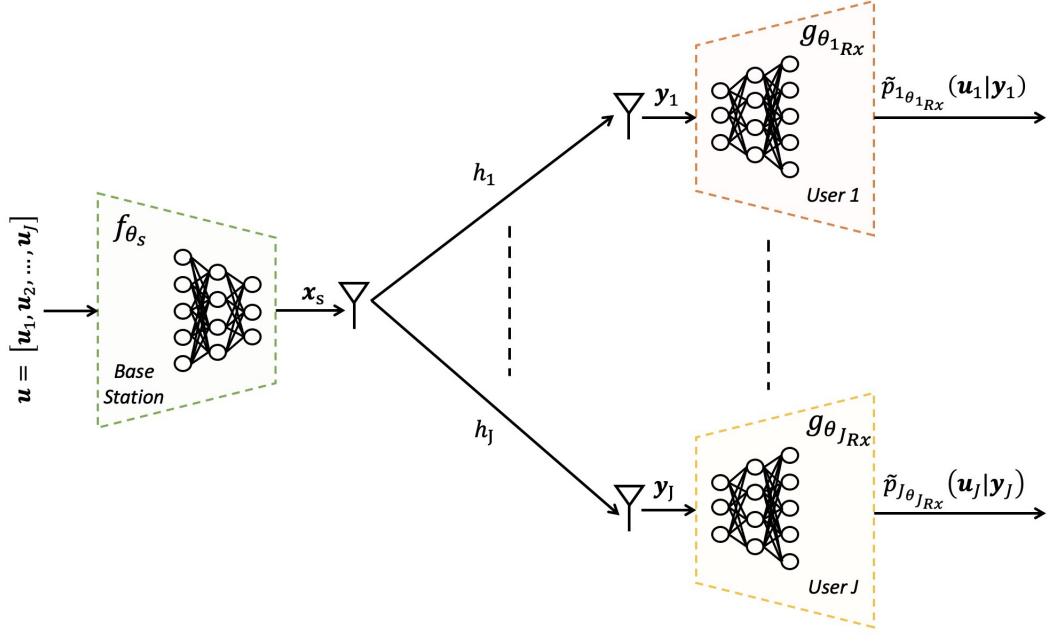


Figure 8.3.1: Proposed AE-based single encoder multiple decoder (SEMD) framework.

M dense layers in the NN encoder, detailed as Definition 6, and given as

$$f_{\theta_s}(\mathbf{u}_s, \mathbf{x}_s) = \mathbf{P}_N(\sigma_M(\mathbf{W}_M \sigma_{M-1}(\mathbf{W}_{M-1} \sigma_{M-2}(\dots \sigma_1(\mathbf{W}_1 \mathbf{u}_s + \mathbf{b}_1) \dots) + \mathbf{b}_{M-1}) + \mathbf{b}_M)) \quad (8.4)$$

where \mathbf{P}_N denotes the power normalization layer, defined as Definition 7, that mandates $\|\mathbf{x}_s\|_2^2 = n$. Now, the BS perform symbol-by-symbol broadcasting to all the J users simultaneously, where the signal received by the j^{th} user is given by (8.2).

Designing of NN-based Multiple Decoders for Downlink Users

Each of the J users have a separate NN-based decoder. Once the n complex baseband symbols are received by the j^{th} downlink user, represented as $\mathbf{y}_j \in \mathbb{C}^n$, it performs block-by-block decoding. In particular, the NN-based decoder at each of the j^{th} downlink user aims to de-map the received n complex symbols to k soft-probabilistic outputs, denoted by $\tilde{p}_{j\theta_{jRx}}(u_j^m | \mathbf{y}_j) \in [0, 1]$, for all $m = \{1, \dots, k\}$. For the sake of clarity, we explicitly define the notation of soft-probabilistic output below

Definition 25 (Soft-probabilistic output notation $\tilde{p}_{j\theta_{jRx}}(u_j^m | \mathbf{y}_j)$) For a given block of n complex baseband symbols \mathbf{y}_j received by the j^{th} user, the probability of decoding the intended bits of the j^{th} user transmitted by the BS, represented as u_j^m (for all m), at the NN decoder of the j^{th} user with the optimization parameters θ_{jRx} , is represented by $\tilde{p}_{j\theta_{jRx}}(u_j^m | \mathbf{y}_j)$, $\forall m$, or, $\tilde{p}_{j\theta_{jRx}}(\mathbf{u}_j | \mathbf{y}_j)$.

Thus, NN-based decoder is jointly performing channel decoding, demodulation design and interference removal of other users' signal. In particular, the decoder designs a NN-based de-mapping function $g_{\theta_{j_{Rx}}}(\mathbf{y}_j, \tilde{p}_{j_{\theta_{j_{Rx}}}(\mathbf{u}_j|\mathbf{y}_j)}) : \mathbf{y}_j \rightarrow \tilde{p}_{j_{\theta_{j_{Rx}}}(\mathbf{u}_j|\mathbf{y}_j)$, where $\theta_{j_{Rx}}$ represents the trainable parameters including weights and bias terms of the NN-based decoder of N dense layers in the decoder of the j^{th} user. Please note the notation $g_{\theta_{j_{Rx}}}(\mathbf{y}_j, \tilde{p}_{j_{\theta_{j_{Rx}}}(\mathbf{u}_j|\mathbf{y}_j))$ represents a decoder de-mapping function g at the j^{th} user, that takes $\mathbf{y}_j \in \mathbb{C}^n$ complex baseband symbols as input to give $\tilde{p}_{j_{\theta_{j_{Rx}}}(\mathbf{u}_j|\mathbf{y}_j) \in [0, 1]^{k_j}$ soft-probabilities as output using the optimization parameter $\theta_{j_{Rx}}$, represented as

$$g_{\theta_{j_{Rx}}}(\mathbf{y}_j, \tilde{p}_{j_{\theta_{j_{Rx}}}(\mathbf{u}_j|\mathbf{y}_j)) = \sigma_N(\mathbf{W}_N \mathcal{B}_N(\sigma_{N-1}(\mathbf{W}_{N-1} \mathcal{B}_N(\sigma_{N-2}(\dots \mathcal{B}_N(\sigma_1(\mathbf{W}_1 \mathcal{B}_N(\mathbf{L}_L(\mathbf{y}_j)) + \mathbf{b}_1) \dots) + \mathbf{b}_{N-1}) + \mathbf{b}_N)) \quad (8.5)$$

where \mathcal{B}_N denotes the batch normalization layer, as given by (8.3), and \mathbf{L}_L denotes the Lambda layer, as detailed in Definition 10. Note the last layer of each j^{th} NN decoder has a *Sigmoid* activation function that outputs soft-probabilities $\tilde{p}_{j_{\theta_{j_{Rx}}}(\mathbf{u}_j^m|\mathbf{y}_j)$, $\forall m$, corresponding to the log-likelihood ratios (LLRs), for $\tilde{p}_{j_{\theta_{j_{Rx}}}(\mathbf{u}_j^m = 1|\mathbf{y}_j) = \sigma(l_j^m)$, as

$$\text{LLR}_j^m := \log \frac{1 - \tilde{p}_{j_{\theta_{j_{Rx}}}(\mathbf{u}_j^m = 0|\mathbf{y}_j)}{\tilde{p}_{j_{\theta_{j_{Rx}}}(\mathbf{u}_j^m = 0|\mathbf{y}_j)} = l_j^m, \quad \forall m \quad (8.6)$$

These LLR can also be employed directly in an outer code of the powerful long block length decoders, such as LDPC and Turbo codes.

8.3.2 Multiple Encoders Multiple Decoders (MEMD)

In this subsection, we detail the AE-based MEMD framework for multi-user downlink network, as shown in Fig. 8.3.2.

Designing of a Multiple NN-based Encoders at the BS

The BS have J separate NN-based encoders for each of the J users. Where the j^{th} NN-based encoder takes the input bit $\mathbf{u}_j \in \{0, 1\}^{k_j}$ of the j^{th} user, and maps it to n complex baseband symbols $\mathbf{x}_s^j \in \mathbb{C}^n$. Thereby, performing joint block channel-coding and modulation design. In particular, the j^{th} encoder designs a NN-based mapping function $f_{\theta_{j_{Tx}}}(\mathbf{u}_j, \mathbf{x}_s^j) : \mathbf{u}_j \rightarrow \mathbf{x}_s^j$, where $\theta_{j_{Tx}}$ represents the optimization parameters including weights and bias terms of the M dense layers in the j^{th} NN encoder, detailed as Defini-

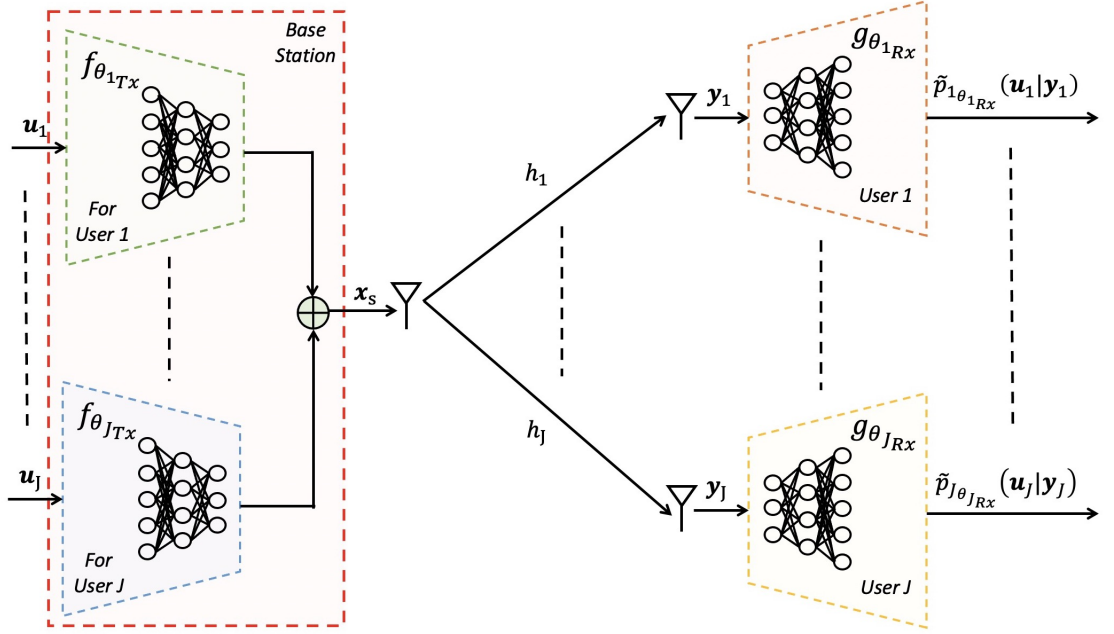


Figure 8.3.2: Proposed AE-based multiple encoder multiple decoder (MEMD) framework.

tion 6, and given as

$$f_{\theta_{jTx}}(\mathbf{u}_j, \mathbf{x}_s^j) = \sigma_M^j(\mathbf{W}_M^j \sigma_{M-1}^j(\mathbf{W}_{M-1}^j \sigma_{M-2}^j(\dots \sigma_1^j(\mathbf{W}_1^j \mathbf{u}_j + \mathbf{b}_1^j) \dots) + \mathbf{b}_{M-1}^j) + \mathbf{b}_M^j) \quad (8.7)$$

where, \mathbf{W}_l^j and \mathbf{b}_l^j denotes the weight and bias terms of the l^{th} dense layer in a j^{th} NN encoder designed for j^{th} downlink user. To superpose these J complex baseband symbols $\mathbf{x}_s^j \in \mathbb{C}^n$ generated by J NN encoders, we simply add all the $2n$ outputs of the J NN-based encoders (without using the PAF) and perform power normalization to obtain superposed n complex baseband symbols $\mathbf{x}_s \in \mathbb{C}^n$ given as

$$\mathbf{x}_s = \mathbf{P}_N(\mathbf{x}_s^1 + \mathbf{x}_s^2 + \dots + \mathbf{x}_s^N) \quad (8.8)$$

where \mathbf{P}_N denotes the power normalization layer, defined as Definition 7, that mandates $\|\mathbf{x}_s\|_2^2 = n$. Now, the BS perform symbol-by-symbol broadcasting to all the J users simultaneously, where the signal received by the j^{th} user is given by (8.2).

Please note the designing of the NN-based multiple decoders for the downlink users remains the same as SEMD (in Section 8.3.1).

8.3.3 Loss Optimization with QoS Constraint

We train the proposed AE-based SEMD and MEMD frameworks by minimizing the total binary cross-entropy (CE) losses for all the J users together. Naturally, while minimizing

the total binary CE loss of all the users together, the users with higher SNR or lower noise variances will easily reconstruct their signals. Thus, the binary CE loss of the users with higher SNR will become zero quickly compared to the users with lower SNR. Hence, NN decoders of the users with higher SNRs will be trained for only high SNR, making them stronger users and vice-versa. To remove the concept of strong and weak users, we introduce a scaling factor that imposes a QoS constraint on the binary CE loss of each user. As a result, the end-to-end training of the proposed AE-based SEMD and MEMD frameworks ensures that each user maintains a similar QoS requirement for decoding its signal. Now, the total binary CE losses for the proposed AE-based SEMD framework ($\mathcal{J}^{\text{SEMD}}$) and MEMD framework ($\mathcal{J}^{\text{MEMD}}$) can be given as follows:

$$\mathcal{J}^{\text{SEMD}}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_{1_{Rx}}, \dots, \boldsymbol{\theta}_{J_{Rx}}) = \sum_{j=1}^J \zeta_j \mathcal{L}_j \left(\mathbf{u}_j, \tilde{p}_{j\boldsymbol{\theta}_{j_{Rx}}}(\mathbf{u}_j | \mathbf{y}_j) \right) \quad (8.9)$$

$$\mathcal{J}^{\text{MEMD}}(\boldsymbol{\theta}_{1_{Tx}}, \dots, \boldsymbol{\theta}_{J_{Tx}}, \boldsymbol{\theta}_{1_{Rx}}, \dots, \boldsymbol{\theta}_{J_{Rx}}) = \sum_{j=1}^J \zeta_j \mathcal{L}_j \left(\mathbf{u}_j, \tilde{p}_{j\boldsymbol{\theta}_{j_{Rx}}}(\mathbf{u}_j | \mathbf{y}_j) \right) \quad (8.10)$$

where the QoS constraint is imposed by ζ_j , which is a scaling factor for the j^{th} downlink user multiplied to the binary CE loss $\mathcal{L}_j(\cdot)$ of the j^{th} downlink user, to make each of the J downlink users have almost similar impact while minimizing the total binary CE loss for the SEMD and MEMD frameworks. We will detail more about updating of this scaling factor in next subsection. Moreover, the notation $\mathcal{L}_j \left(\mathbf{u}_j, \tilde{p}_{j\boldsymbol{\theta}_{j_{Rx}}}(\mathbf{u}_j | \mathbf{y}_j) \right)$ represents the binary CE loss term, as defined in Definition 13, for the j^{th} downlink user's input bits \mathbf{u}_j at the NN encoder and soft probabilistic output $\tilde{p}_{j\boldsymbol{\theta}_{j_{Rx}}}(\mathbf{u}_j | \mathbf{y}_j)$ at its NN decoder.

8.3.4 Training and Testing Procedure

Now, we can estimate the total binary CE loss for the AE-based SEMD and MEMD frameworks in (8.9) and (8.10) via mini-batch sampling [41] as follows:

$$\mathcal{J}^{\text{SEMD}}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_{1_{Rx}}, \dots, \boldsymbol{\theta}_{J_{Rx}}) = \frac{1}{B^s} \sum_{q=1}^Q \sum_{b^s=1}^{B^s} \sum_{j=1}^J \sum_{m=1}^k \zeta_j \mathcal{L}_j \left(u_j^{(q \times b^s, m)}, \tilde{p}_{j\boldsymbol{\theta}_{j_{Rx}}}(u_j^{(q \times b^s, m)} | \mathbf{y}_j^{(q \times b^s)}) \right) \quad (8.11)$$

$$\mathcal{J}^{\text{MEMD}}(\boldsymbol{\theta}_{1_{Tx}}, \dots, \boldsymbol{\theta}_{J_{Tx}}, \boldsymbol{\theta}_{1_{Rx}}, \dots, \boldsymbol{\theta}_{J_{Rx}}) = \frac{1}{B^s} \sum_{q=1}^Q \sum_{b^s=1}^{B^s} \sum_{j=1}^J \sum_{m=1}^k \zeta_j \mathcal{L}_j \left(u_j^{(q \times b^s, m)}, \tilde{p}_{j\boldsymbol{\theta}_{j_{Rx}}}(u_j^{(q \times b^s, m)} | \mathbf{y}_j^{(q \times b^s)}) \right) \quad (8.12)$$

where B denotes the batch size and $Q = \frac{S_{\text{train}}}{B}$, where S_{train} denotes the total number of training samples. The weights and bias terms in the NN of the AE frameworks are

updated using the stochastic gradient descent (SGD) method using back-propagation [41]. Let us denote the NN optimization parameters for the AE-based SEMD framework as $\Theta = \{\theta_s, \theta_{1_{Rx}}, \dots, \theta_{J_{Rx}}\}$ and similarly the NN optimization parameters for the AE-based MEMD framework can be represented as $\Theta = \{\theta_{1_{Tx}}, \dots, \theta_{J_{Tx}}, \theta_{1_{Rx}}, \dots, \theta_{J_{Rx}}\}$, then we update the weights using the Adam optimizer as detailed in Sec. 2.5.1.

Please note in (8.9) and (8.10), we impose the QoS constraint on each of the J users using the scaling factor ζ_j , with the aim that the end-to-end training makes all the user's have equal importance in decoding their respective signals correctly. Thus, we propose to update the scaling factor for each of the j^{th} user after each epoch t . Where the scaling factor ζ_j for the epoch (t) can be defined as the binary CE loss of the j^{th} user at epoch ($t - 1$) divided by the total binary CE loss of all the users at epoch ($t - 1$), respectively. Now, we give the update rule for the scaling factor for the t^{th} epoch below

$$\zeta_j^{(t)} = \frac{\mathcal{L}_j^{(t-1)}(\mathbf{u}_j, \tilde{p}_{j\theta_{j_{Rx}}}(\mathbf{u}_j|\mathbf{y}_j))}{\sum_{j=1}^J \mathcal{L}_j^{(t-1)}(\mathbf{u}_j, \tilde{p}_{j\theta_{j_{Rx}}}(\mathbf{u}_j|\mathbf{y}_j))}, \quad \forall j, t \quad (8.13)$$

Please note that for the first epoch at $t = 1$, we initialize the scaling factor as $\zeta_j^{(t=1)} = 1/J$ for all $j = \{1, \dots, J\}$. This means that we impose a similar QoS requirement even in the first epoch. Let us consider a simple two-user downlink scenario and study the training process of the proposed AE framework, suppose at $t = 5^{\text{th}}$ epoch, the binary CE loss for a $j = 1$ user becomes 0.5 and $j = 2$ user becomes 0.1. Then after this epoch, we update the scaling factor using (8.13) which becomes $\zeta_1 = 0.1/0.6 = 0.167$ for $j = 1$ user and $\zeta_2 = 0.5/0.6 = 0.833$ for $j = 2$ user. Thus using (8.9), (8.10), in the next epoch, i.e. $t = 6^{\text{th}}$ epoch, the minimization of the binary CE loss of the $j = 2$ user will be given more importance compared to the $j = 1$ user. Hence, following the update rule for scaling factor in (8.13), at each epoch, we evaluate the binary CE losses for each user and give more importance to minimizing the binary CE loss of the user with higher binary CE losses in the next epoch. Hence, with each training epoch, the AE framework tries to make the binary CE loss of each user equally minimum, thereby inherently imposing the same QoS constraint for all the J downlink users. Please note that we can also simply modify the QoS constraint to impose strong and weak user constraint by multiplying a user's binary CE loss with an additional factor.

For the trained AE-based SEMD and MEMD frameworks, we perform predictions on the testing set to obtain soft probabilistic outputs $\tilde{p}_{j\theta_{j_{Rx}}}(\mathbf{u}_j|\mathbf{y}_j)$ for all the J users. Similar to Section 2.5.2, we then perform hard decision decoding by keeping a threshold of 0.5 to obtain the decoded bits of each downlink user.

Table 8.3.1: NN Architectures for AE-based Modulation designs.

DL Network Layout For		Input layer	Layer-1	Layer-2	Layer-3	Layer-4	Layer-5
SEMD	Encoder	$\{k_1, \dots, k_J\}$	512 (L)	256 (L)	64 (L)	2 (L)	2 (PN)
MEMD	Encoder _{<i>j</i>}	k_j	512 (L)	256 (L)	64 (L)	2 _{<i>j</i>} (L)	–
	Symbol Add + PN	$\{2_1, \dots, 2_J\}$	2 (\mathcal{A}_D)	2 (PN)	–	–	–
SEMD, MEMD	Decoder _{<i>j</i>}	2	2 (\mathcal{C}_D) + \mathcal{B}_N	256 (L) + \mathcal{B}_N	128 (R) + \mathcal{B}_N	32 (R) + \mathcal{B}_N	k_j (S)

8.3.5 AE-based (SEMD and MEMD Frameworks) Modulation, BCM and d-BCM Designs

In this work, we focus on fading channels and propose AE-based SEMD and MEMD frameworks-based modulation, BCM, and d-BCM designs, by designing the Lambda layers (\mathbf{L}_L) in each of the j^{th} NN-based decoder, as follows:

- *Modulation Design* – Herein, we assume perfect CSI knowledge of all the users and perform channel equalization of each user’s channel h_j in the Lambda layer.
- *BCM Design* – Herein, we assume perfect CSI knowledge of all the users and perform channel equalization of each user’s channel h_j in the Lambda layer.
- *d-BCM Design* – Herein, we assume we do not have any form of CSI knowledge of any of the J users. Unlike the previous chapters 2–5, where we employ a radio transformer network (RTN) to estimate and remove the fading channel. Herein, similar to chapter 6, we do not consider any RTN in the Lambda Layer of the NN decoder of any user. Thus, the n received complex baseband symbols directly become the NN decoder’s input, without any Lambda layer-based pre-processing.

8.3.6 NN Architectures of the AE-based SEMD and MEMD Frameworks

We provide the NN architectures for the proposed AE-based SEMD and MEMD frameworks in Table 8.3.1 for modulation design and in Table 8.3.2 for BCM and d-BCM design. Please note in Tables 8.3.1, 8.3.2, the notation $X(Y)$ denotes X number of neurons in the layer. The $Y = L, R, S$ indicates a dense layer is employed, where $Y = L, R, S$ denotes the Linear, ReLU, and Sigmoid activation functions employed in the dense layer, respectively. The $Y = \text{PN}, \mathcal{A}_D, \mathcal{C}_D$ indicates a Lambda layer \mathbf{L}_L is employed, where PN denotes power normalization layer, \mathcal{C}_D denotes complex channel divide, and \mathcal{A}_D denotes symbol addition. Also, \mathcal{B}_N denotes the batch normalization layer as defined in (8.3). Please note, every layer in the NN decoder of each of the J downlink users is followed by a \mathcal{B}_N layer, as shown in (8.5). We now elaborate the advantages of the proposed NN architecture below

Table 8.3.2: NN Architectures for AE-based BCM and d-BCM designs.

DL Network Layout For	Input layer	Layer-1	Layer-2	Layer-3	Layer-4	Layer-5	Layer-6	
SEMD	Encoder	$\{k_1, \dots, k_J\}$	1024 (L)	512 (L)	256 (L)	64 (L)	$2n$ (L)	$2n$ (PN)
MEMD	Encoder _{<i>j</i>}	k_j	1024 (L)	512 (L)	256 (L)	64 (L)	$2n_j$ (L)	-
	Symbol Add + PN	$\{2n_1, \dots, 2n_J\}$	$2n$ ($\mathcal{A}_{\mathcal{D}}$)	$2n$ (PN)	-	-	-	-
SEMD & MEMD	Decoder _{<i>j</i>} (for d-BCM)	$2n$	512 (L) + $\mathcal{B}_{\mathcal{N}}$	256 (R) + $\mathcal{B}_{\mathcal{N}}$	128 (R) + $\mathcal{B}_{\mathcal{N}}$	64 (R) + $\mathcal{B}_{\mathcal{N}}$	k_j (S)	-
	Decoder _{<i>j</i>} (for BCM)	$2n$	$2n$ ($\mathcal{C}_{\mathcal{D}}$) + $\mathcal{B}_{\mathcal{N}}$	512 (L) + $\mathcal{B}_{\mathcal{N}}$	256 (R) + $\mathcal{B}_{\mathcal{N}}$	128 (R) + $\mathcal{B}_{\mathcal{N}}$	64 (R) + $\mathcal{B}_{\mathcal{N}}$	k_j (S)

- We employ a Fixed bit-wise AE framework, offering the advantages of bit-wise AE and Fixed AE frameworks, detailed in Chapter 2.
- We create a generalizable NN architecture for the decoders at the downlink users, that remains the same for all the downlink users, in both the SEMD and MEMD frameworks. The benefits of this is two folds. Firstly, the same NN architecture of the decoder can be directly employed to support the increasing number of multiplexed users in the proposed frameworks. Secondly, since each user has same decoding prowess (because of same number of optimization parameters), thus we remove the concept of stronger (near) user with high decoding complexity because of multiple SICs or weaker (far) with low decoding complexity because of no or minimum SICs. As a result, we are able to create an AE framework that can perform well for varying SNR gaps (ΔSNR) between the users, even for the case when $\Delta\text{SNR} = 0$ dB, i.e., all the users have the same receiving SNR.
- We create a generalizable NN architecture of the encoder that remains the same in both the SEMD and MEMD frameworks. Thus at the BS, the SEMD framework has J times lesser complexity than the MEMD framework.

8.3.7 Hyper-Parameter Settings

We consider $S_{\text{train}} = 3 \times 10^5$ training samples, $S_{\text{test}} = 10^5$ testing samples. For AE-based modulation design we generate S_{train} samples at the SNR in the set $\mathcal{S} = \{5\}$ dB, and for AE-based BCM and d-BCM designs we generate S_{train} samples at the SNRs in the set $\mathcal{S} = \{3, 23, 28\}$ dB. Thus, we observe that training on a single SNR point is sufficient for modulation designs. However, we need multiple SNR points to train the BCM and d-BCM designs. This may be because we are designing the AE-based modulations in 2-dimensional space while designing the AE-based BCM and d-BCM designs in $2n$ -dimensional space. Further, the AE-based SEMD and MEMD frameworks are implemented in Keras [43] with TensorFlow [44] as backend. For training, we utilize SGD with Adam optimizer [42], where the weights are initialized using Glorot initializer [45]. We keep the training epochs (E) as 50. For the AE-based modulation design, we keep the

learning rate as $\tau = 0.000125$. In contrast for the AE-based BCM and d-BCM designs, we keep the learning rate as $\tau = 0.00125$. We also utilize early stopping; in particular, we monitor the validation CE loss and keep the patience as 20 and minimum change in the loss as 0.075 while training the proposed AE frameworks.

8.3.8 Advantages of the proposed AE framework over NOMA

We now summarize the advantages of the proposed AE-based SEMD and MEMD frameworks for multi-user downlink networks over the conventional NOMA networks below

- In conventional NOMA, each user informs the BS of its channel gain, and the BS then optimizes the PAF depending upon the channel gains to superpose the symbols, whose complexity increases exponentially with the number of users in the network. With the proposed AE frameworks, we remove the utilization of PAF at BS while operating under i.i.d. Rayleigh fading channels. Thereby removing the need to feedback channel gains to the BS and optimizing the PAF, thus saving multiple time-slots (increasing the sum rate) and reducing the complexity (required for PAF optimization).
- In conventional NOMA, the BS needs to inform the users of the PAF and SIC (or user) ordering. With proposed AE frameworks, we remove SIC from all the users. Thus, removing the necessity to transmit the PAF and SIC ordering information to the users, saving multiple time-slots (increasing the sum rate). Moreover, we remove the concept of strong or weak users. Thus we can decode the users even on the same receiving SNR, thereby making our design methodology independent of the SNR gap (ΔSNR) between the multiplexed users.
- In conventional NOMA, as the number of multiplexed users increases, the total number of SICs performed increases exponentially. Notably, we need to perform $\frac{J(J-1)}{2}$ SICs to decode the signals of all the J multiplexed users. Therefore, the chances of error propagation and decoding complexity increase for the stronger users. With the proposed AE frameworks, we remove the necessity of SIC. Thus all the users have the same decoding complexity, and the error propagation due to multiple SICs is also removed.
- In the proposed AE framework, we impose an equal QoS constraint on each of the J users, such that the BER performance of each user depends on the receiving SNR only.
- Most importantly, we proposed AE-based d-BCM designs which can decode the signals without the CSI knowledge. In contrast, conventional NOMA can not decode the signals without the CSI knowledge.

8.4 Performance Evaluation

In this section, we analyze the performance of the proposed AE-based SEMD and MEMD frameworks performing modulation design and BCM or d-BCM design for multi-user downlink networks. We consider an RBF channel and assume that the channel remains constant for n symbol transmissions and changes thereafter. We consider the conventional downlink PD-NOMA as a benchmark. We consider $J = \{2, 3, 4\}$ in this section.

8.4.1 AE-based Modulation Design

In this subsection, we analyze the performance of the proposed AE-based SEMD and MEMD frameworks performing the modulation designs by keeping $n = 1$ for multi-user downlink networks. We consider the conventional downlink PD-NOMA utilizing the maximum likelihood detector (MLD) as a benchmark. We operate under i.i.d. fast fading Rayleigh channels, i.e., we have a different complex channel coefficient for each transmitted symbol. Moreover, we keep $(n, k_j) = (1, 1)$ for all $j = \{1, \dots, J\}$. For the sake of fair comparison, we consider the presence of the CSI knowledge for the conventional and proposed AE frameworks.

Traditional Modulation Designs – Signal Superposition

Since $k = 1$, we utilize BPSK modulation for conventional NOMA. As shown in Fig. 8.4.1, we consider two cases, (1) *without rotation* - here all the symbols are on the I -axis only, before superposition and after superposition, and (2) *rotated* - here we rotate the symbols of the users before superposition, in particular, each user is rotated by an additional $90^\circ, 60^\circ, 45^\circ$ for two, three and four user case, before superposition as shown in Fig. 8.4.1d-8.4.1f respectively. We superpose the symbols of the user by utilizing the optimal PAF (α_j^P), and after superposition we get the superposed symbols as shown in Fig. 8.4.1a-8.4.1c for the *without rotation* scenario. Also, after superposition of the rotated symbols in Fig. 8.4.1d-8.4.1f using the PAF we get the superposed symbols as shown in Fig. 8.4.1g-8.4.1i for the *rotated* scenario. Clearly, optimal rotation becomes difficult as the number of users increases, such that after superposition it leads to maximization of the minimum Euclidean distance, as seen in Fig. 8.4.1i.

AE-based Modulation Designs – Signal Superposition

Now, we evaluate the constellation designed by the proposed AE-based SEMD and MEMD frameworks in Fig. 8.4.2. To obtain the constellation points, we input the bits to the encoder at the BS, and obtain the complex baseband symbols as output. The constellation obtained by the proposed AE frameworks are rotated for two, three and four users, thereby

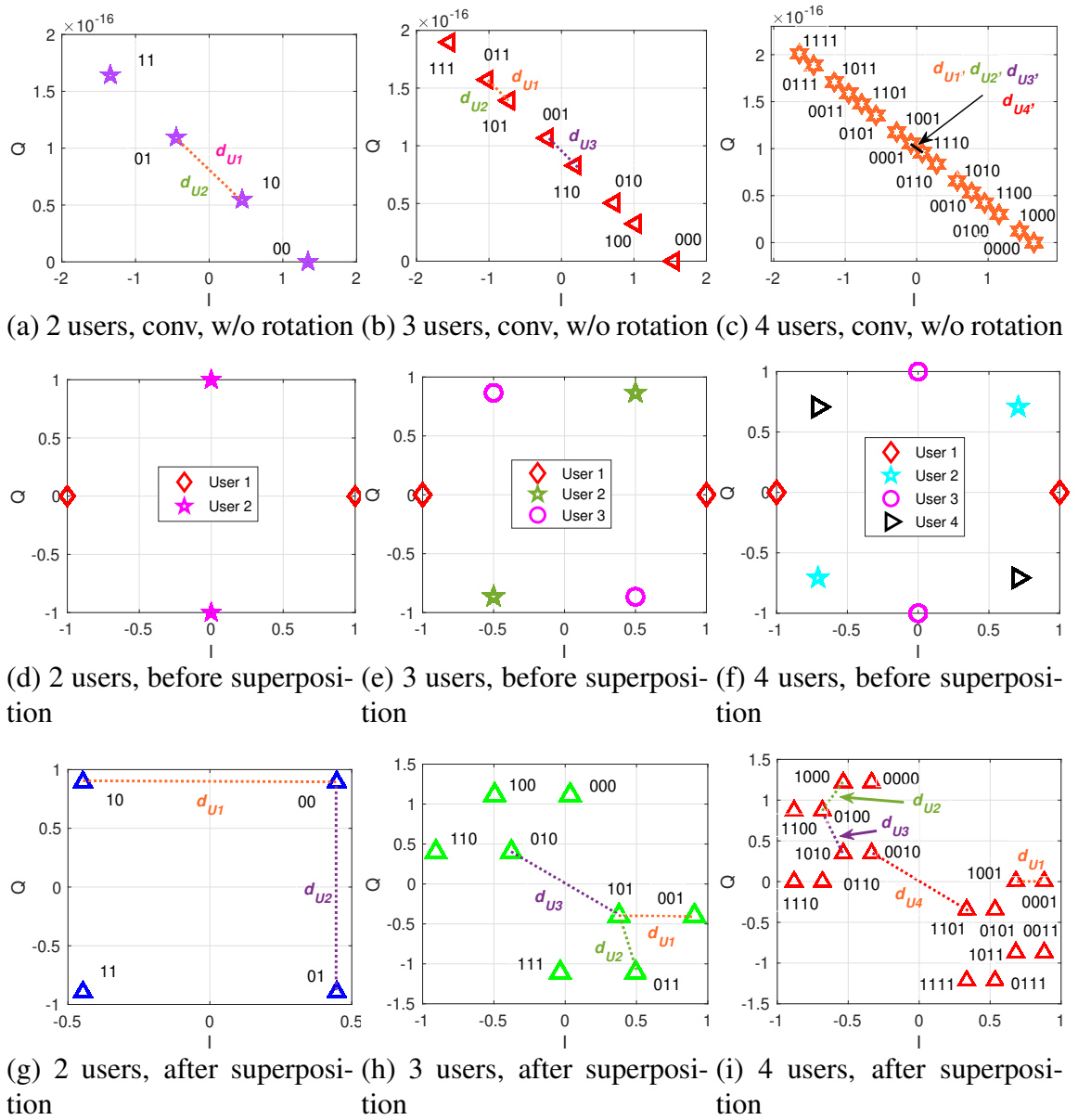


Figure 8.4.1: Constellation mapping for conventional and rotated NOMA $(n, k_j) = (1, 1)$

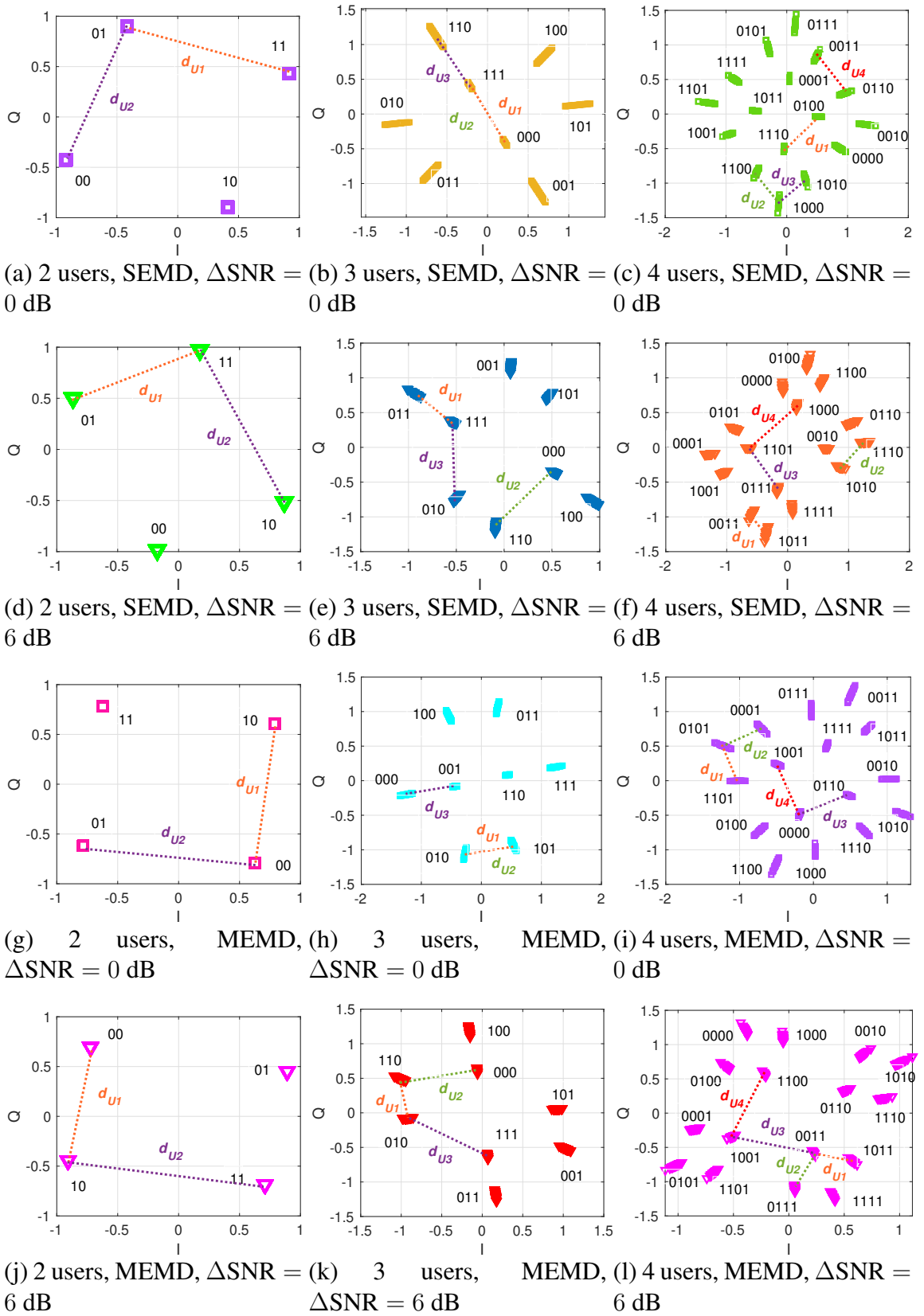


Figure 8.4.2: Constellation mapping obtained for proposed SEMD & MEMD models $(n, k_j) = (1, 1)$

Table 8.4.1: Minimum squared Euclidean distance of each user $(n, k_j) = (1, 1)$

No. of users	User index	Conventional	Rotated	SEMD		MEMD	
				$\Delta\text{SNR} = 0 \text{ dB}$	$\Delta\text{SNR} = 6 \text{ dB}$	$\Delta\text{SNR} = 0 \text{ dB}$	$\Delta\text{SNR} = 6 \text{ dB}$
2	$d_{\min,U1}^2$	0.80	0.80	1.98	1.31	1.98	1.33
	$d_{\min,U2}^2$	0.80	3.19	2.01	2.70	2.01	2.67
3	$d_{\min,U1}^2$	0.087	0.279	0.866	0.328	0.672	0.329
	$d_{\min,U2}^2$	0.087	0.523	0.866	0.975	0.672	0.904
	$d_{\min,U3}^2$	0.151	1.203	0.645	1.186	0.649	1.269
4	$d_{\min,U1}^2$	0.025	0.040	0.504	0.123	0.272	0.124
	$d_{\min,U2}^2$	0.025	1.141	0.209	0.271	0.321	0.277
	$d_{\min,U3}^2$	0.025	0.297	0.232	0.549	0.478	0.589
	$d_{\min,U4}^2$	0.025	0.926	0.506	0.957	0.564	0.871

increasing the minimum Euclidean distance compared to the conventional (non-rotated) and rotated constellation in Fig. 8.4.1a-8.4.1c and Fig. 8.4.1g-8.4.1i. Moreover, most of the symbols in the obtained constellation are automatically arranged in a Gray-coded format¹. Furthermore, for three users scenario, when the SNR gap between the users ($\Delta\text{SNR} = 0 \text{ dB}$) becomes same, the symbols are arranged in a circular format. Whereas for the four users scenario, a rotated 16-QAM like constellation is designed by the proposed AE frameworks. Also, in three and four users scenario, multiple symbols are designed with similar I-Q values for the same bit mapping. This is because we have an extra degree of freedom to elongate the symbols, possibly leading to constructive interference at the users.

For the constellation obtained for the four users in Fig. 8.4.2c, Fig. 8.4.2f, Fig. 8.4.2i and Fig. 8.4.2l, we can see that the two least significant bits for any symbol in a quadrant is same, except for the SEMD at $\Delta\text{SNR} = 0 \text{ dB}$. This denotes that for SEMD ($\Delta\text{SNR} = 0 \text{ dB}$) and MEMD each quadrant is describing the third and fourth users and within each quadrant we have first and second user.

Now, we calculate the square of minimum Euclidean distance, defined in Definition 16, between each superposed user in Table 8.4.1 with the aim to analyze and compare the user fairness achieved by conventional NOMA and proposed AE frameworks. We denote these distances in Fig. 8.4.1, 8.4.2 with $d_{U1}, d_{U2}, d_{U3}, d_{U4}$ denoting the symbols where the first, second, third and fourth user will have minimum squared Euclidean distance (MSED), respectively.

Remark 30 *Let us consider the two-user case in Table 8.4.1. The MSED for the superposition of the conventional constellation is small for both the users. In contrast, the superposition of the rotated constellation leads to increased MSED for the second user. In contrast, the first user has the same MSED as without rotation scenario. However, for the AE-based SEMD and MEMD frameworks, the MSED between two users is comparable.*

¹However, we need to investigate further why some symbols are not arranged in gray-coded format and are left for future works.

Whereas when the SNR gap between the users is the same, i.e., $\Delta\text{SNR} = 0$ dB, then the MSED is almost exactly the same. This shows that we are achieving exceptional user fairness compared to the conventional modulation rotations etc. Interestingly, as the SNR gap between the users' increases, i.e., $\Delta\text{SNR} = 6$ dB, then the user with higher receiving SNR (second user in this case) will have a higher MSED compared to the first user because we are designing the constellation by training the AE framework in an end-to-end manner by minimizing the total binary CE loss as shown in (8.9), (8.10). Using the QoS constraint ensures fairness in optimizing the binary CE loss for both users. However, AWGN at the user becomes a limiting factor in improving the constellations after a certain point. Thus, the AE framework is deciding to place the user's symbols with higher receiving SNR as far apart as possible to decode it more accurately. A similar trend can be seen for three and four users in Table 8.4.1.

BER Evaluation

We evaluate the BER performance of the PD-NOMA using the conventional and rotated modulation designs and the proposed AE-based SEMD and MEMD frameworks in Fig. 8.4.3, 8.4.4. As the SNR increases, the BER reduces. Moreover, both the proposed AE-based SEMD and MEMD frameworks outperform the PD-NOMA for varying SNR gaps and the number of users. We summarize these performance gains at the $E_b/N_0 = 10$ dB for farthest user in Table 8.4.2. These BER performance gains can be explained by the Remark 31 below.

Remark 31 *The end-to-end learning-based AE design leads to optimal signal superposition with optimal constellation rotation and translation at the BS. This is because we are minimizing the binary CE loss, which is a better metric. After all, it maximizes the bit-wise MI or the generalized MI (detailed in Chapter 2), thereby directly maximizing the MSED between superposed symbols at the BS. Also, we remove the SIC from the stronger users; thus, the error propagation due to multiple SICs at the stronger users is removed.*

In Fig. 8.4.3, 8.4.4 and Table 8.4.2, we can see that the proposed AE frameworks achieve higher BER performance gains with reducing SNR gap (ΔSNR) between the users, and this can be explained by the Remark 32 below.

Remark 32 *For $\Delta\text{SNR} = 0$ dB, i.e., all the users are on same receiving SNR, the optimal PAF (α_j^P) will be $1/J$. Thus, in this case, the signal interference at each user will become very high because of the other users, making decoding the signals at any of the downlink users almost impossible. Thus we consider unequal PAF for the users. However still, this lowers the decoding capability of all the users equally because SICs is not performing well. In contrast, the proposed AE-based framework design has a QoS constraint for all*

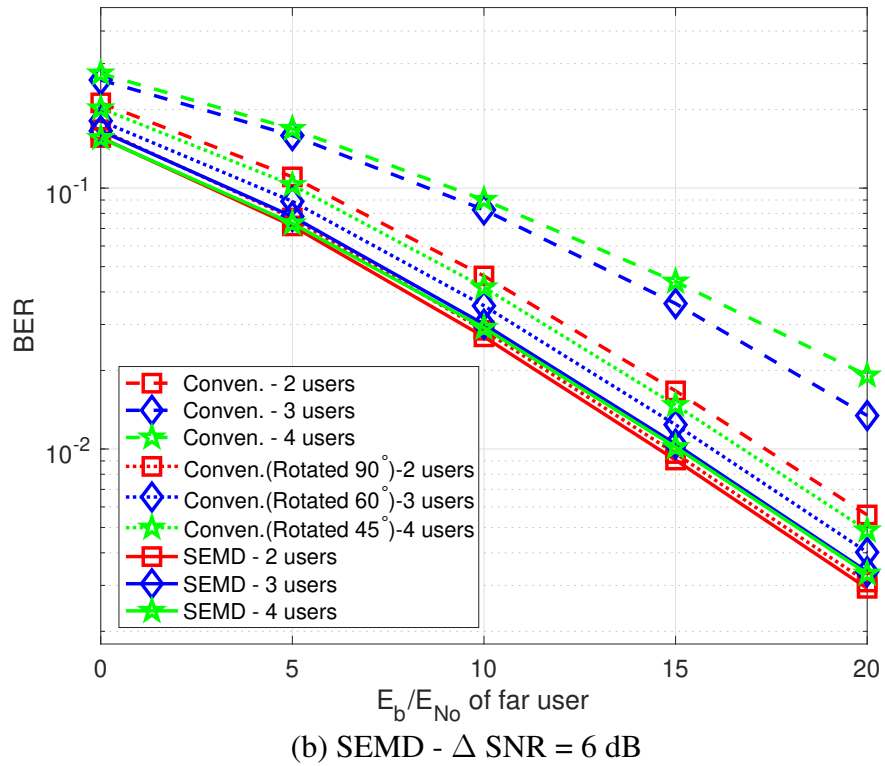
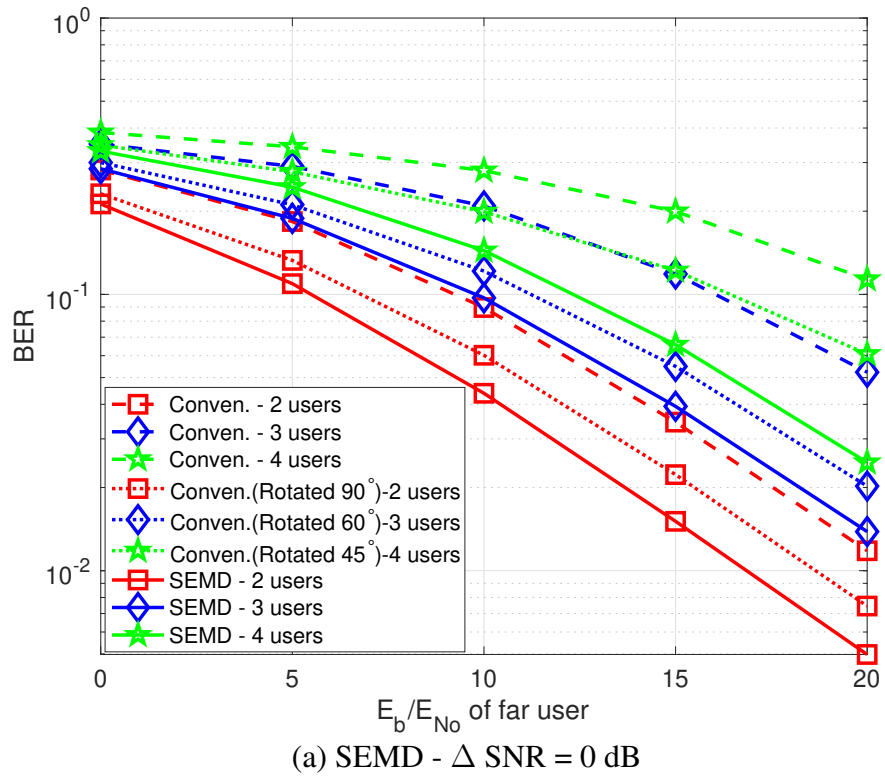
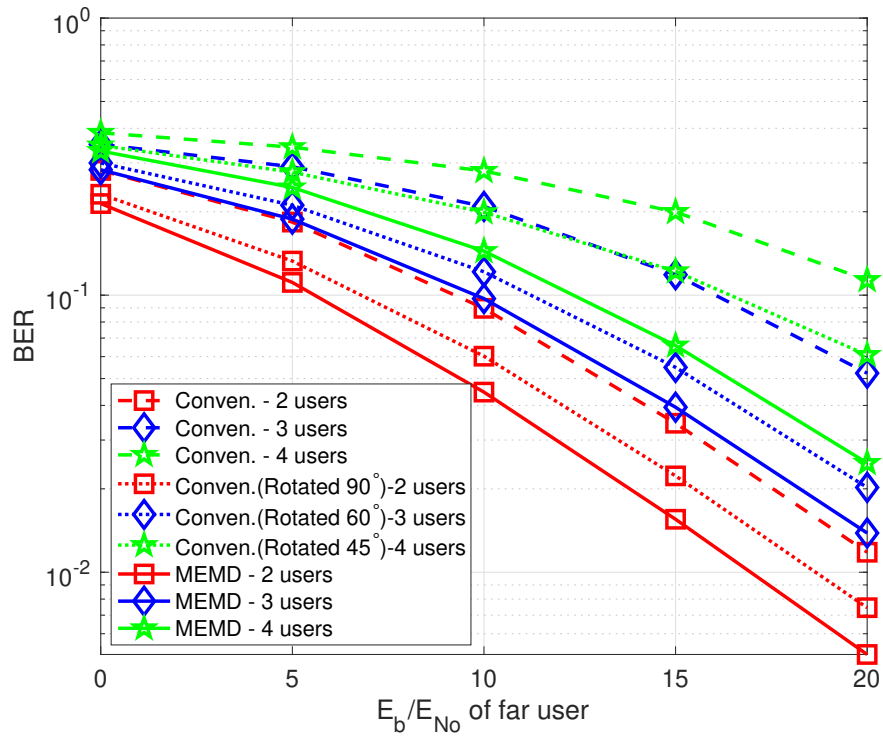
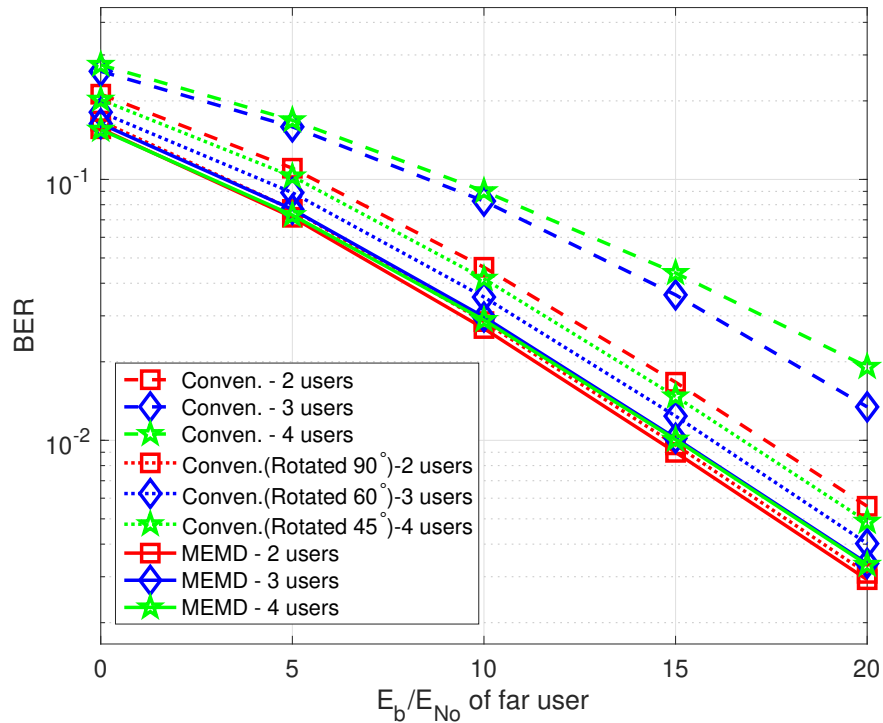


Figure 8.4.3: Performance evaluation for AE-based SEMD framework performing modulation designs with $(n, k_j) = (1, 1)$.



(a) MEMD - Δ SNR = 0 dB



(b) MEMD - Δ SNR = 6 dB

Figure 8.4.4: Performance evaluation for AE-based MEMD framework performing modulation designs with $(n, k_j) = (1, 1)$

Table 8.4.2: Performance gains of the proposed AE-based SEMD and MEMD frameworks performing modulation designs over the PD-NOMA with conventional and rotated modulation designs, for $E_b/N_0 = 10$ dB at the farthest user.

No. of downlink users	SEMD				MEMD			
	$\Delta\text{SNR} = 0$ dB		$\Delta\text{SNR} = 6$ dB		$\Delta\text{SNR} = 0$ dB		$\Delta\text{SNR} = 6$ dB	
	Conv.	Rotated	Conv.	Rotated	Conv.	Rotated	Conv.	Rotated
2	4 dB	1.8 dB	2.8 dB	0.5 dB	3.9 dB	1.8 dB	2.7 dB	0.2 dB
3	6.2 dB	1.7 dB	6 dB	1 dB	6.1 dB	1.7 dB	5.9 dB	0.9 dB
4	8 dB	3.5 dB	7 dB	2 dB	8 dB	3.5 dB	7.4 dB	1.9 dB

users. No user is treated as a stronger or weaker user. Thus the AE framework aims to minimize the BER of all the users equally by designing optimal constellations. Moreover, when the SNR gap increases $\Delta\text{SNR} = 6$ dB, then in the conventional methods, SICs at stronger users are minimally affected. Thus the conventional methods are also able to decode the signals properly. Thus, we see lower performance gains with an increasing SNR gap between the users.

In Fig. 8.4.3, 8.4.4 and Table 8.4.2, the proposed AE frameworks achieve higher BER performance gains in comparison to NOMA with conventional modulation design, as compared to NOMA with rotated modulation design. This is because the MSED between the superposed symbols increases in the rotated modulation designs, as shown in Table 8.4.1, leading to better BER performance compared to conventional modulation designs.

In Fig. 8.4.3, 8.4.4 and Table 8.4.2, we see that as the number of users increases the BER performance gains of the proposed AE frameworks over the NOMA increases, explained by the Remark 33 below.

Remark 33 *In conventional PD-NOMA, multiple SICs are implemented at the stronger users; in fact, the total number of SICs performed in the PD-NOMA is $\frac{J(J-1)}{2}$. Thereby, the chances of error propagation increase exponentially with an increasing number of users. In contrast, each user decodes its signal directly in the proposed AE frameworks without performing any SIC.*

In Table 8.4.2, we can see that both the AE-based SEMD and MEMD frameworks perform equally well for varying SNR gaps and increasing number of users. This is because both the proposed frameworks can design optimal constellations (shown in Fig. 8.4.2), which leads to almost the same minimum Euclidean distance between the multiplexed users (as shown in Table 8.4.1), leading to similar BER performance gains.

Achieved user fairness

In Fig. 8.4.5, we analyze the user fairness achieved by the proposed AE-based SEMD and MEMD frameworks and conventional PD-NOMA using conventional and rotated

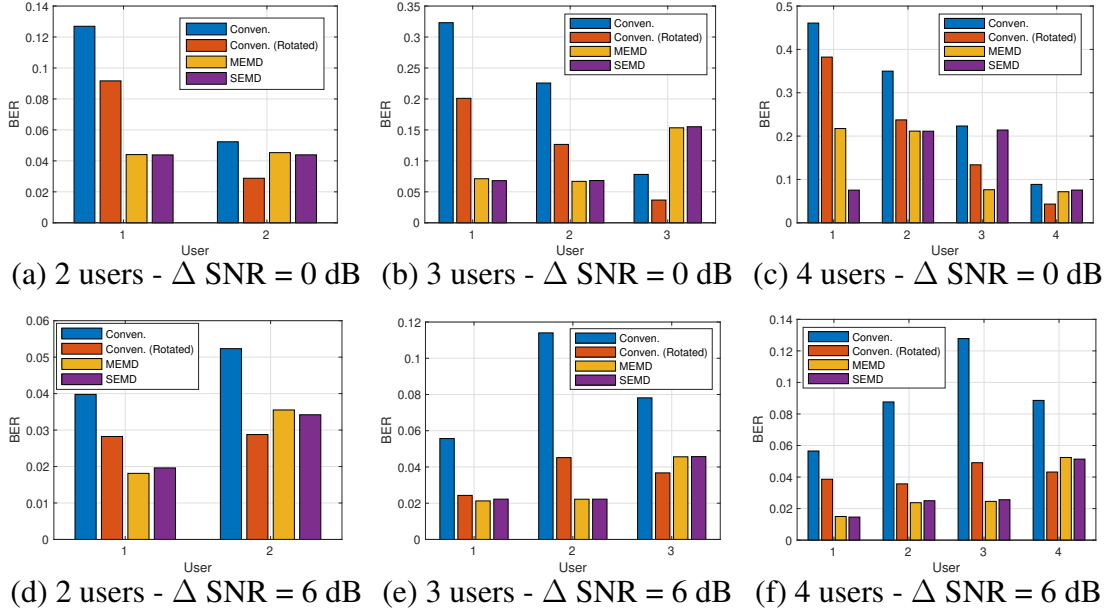


Figure 8.4.5: BER of each downlink user for $E_b/N_0 = 10$ dB at the farthest user for $(n, k_j) = (1, 1)$ (modulation designs).

modulation designs for $E_b/N_0 = 10$ dB at the farthest user. For users at the same receiving SNR, i.e., Δ SNR = 0 dB, we observe smaller BER for weaker users compared to the stronger user in the conventional NOMA. The optimal PAF (α_j^P) should be $1/J$, but if we use this PAF, the stronger users can not perform the SICs. Thus, we gave the highest power to the J^{th} user and the least power to the 1^{st} user while superposing the signal. Thereby, it becomes easier for the J^{th} user to decode its signal compared to the 1^{st} user, which also has the burden to implement $J - 1$ SICs. Moreover, when the SNR gap between the users is high, i.e., Δ SNR = 6 dB, the conventional NOMA tries to achieve user fairness because of the chosen PAF values. Moreover, user fairness is almost achieved in rotated NOMA because of the optimal constellation design due to the chosen PAF. Also, a higher SNR gap between the users allows proper implementation of SICs at the stronger users. Again, this can be understood with the MSED achieved between multiplexed users for all the schemes detailed in Table 8.4.1 and Remark 30, respectively.

Moreover, in Fig. 8.4.5, the AE-based SEMD and MEMD frameworks completely achieve user fairness. This is because of the included QoS constraint with the scaling factor ζ_j while minimizing the total binary CE loss. The scaling factor scales the j^{th} user's binary CE loss at each epoch, depending on the binary CE losses of all the other users. In other words, with each training epoch, the AE-based SEMD and MEMD frameworks try to make the scaling factor $\zeta_j \approx 1/N$ such that each user is given equal importance in decoding the signal. Thus, we can also see that when the SNR gap between the users' increases, i.e., Δ SNR = 6 dB, all the users except for the farthest user approximately have the same BER, while the last user's BER is slightly more than that of other user's

because of the higher noise levels compared to the nearer users.

8.4.2 AE-based BCM and d-BCM Designs

In this subsection, we analyze the performance of the proposed AE-based SEMD and MEMD frameworks performing the BCM and d-BCM designs by keeping $n = 7$ for multi-user downlink networks. We consider the conventional downlink PD-NOMA is utilizing the BPSK or d-BPSK modulation and $(7, 4)$ Hamming code with the MLD as a benchmark. We operate under i.i.d. RBF channels, such that the channel remains constant for $n = 7$ transmissions. Moreover, we keep $(n, k_j) = (7, 4)$.

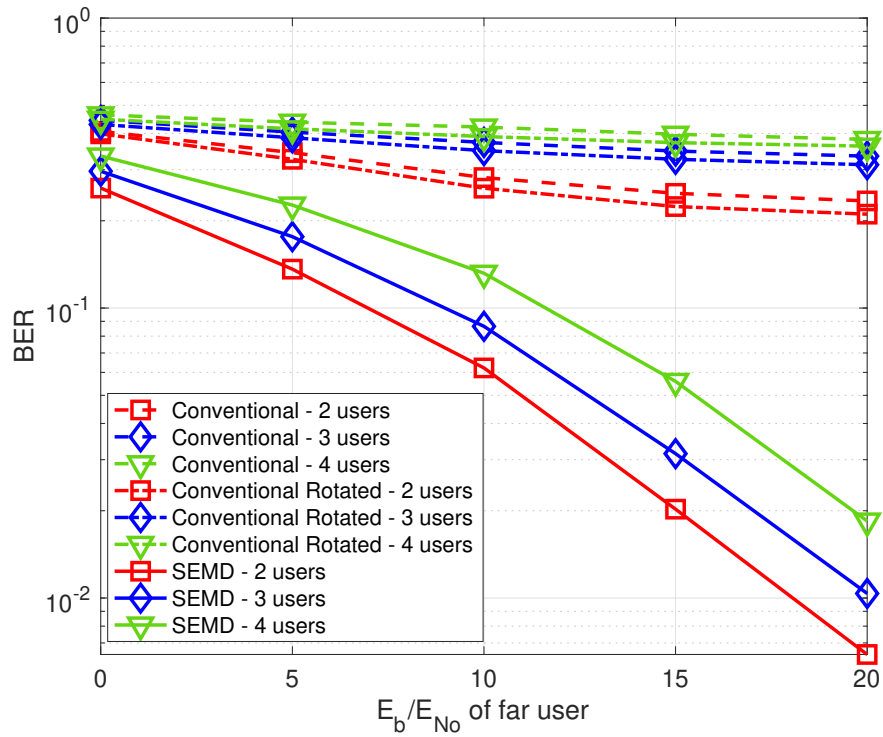
AE-based d-BCM Design

In Fig. 8.4.6, 8.4.7, we can see that the PD-NOMA is unable to decode the signals. This is because we require the CSI knowledge to implement the SIC. Thus we can not decode the signals at the stronger users (implementing multiple SICs) and only decode the signal of the farthest user that does not implement SIC by considering other users' signals as noise. Thus, the average BER for all the users increases with the increasing number of multiplexed users. Moreover, there is a slight improvement in the BER performance with rotated modulations over conventional modulations.

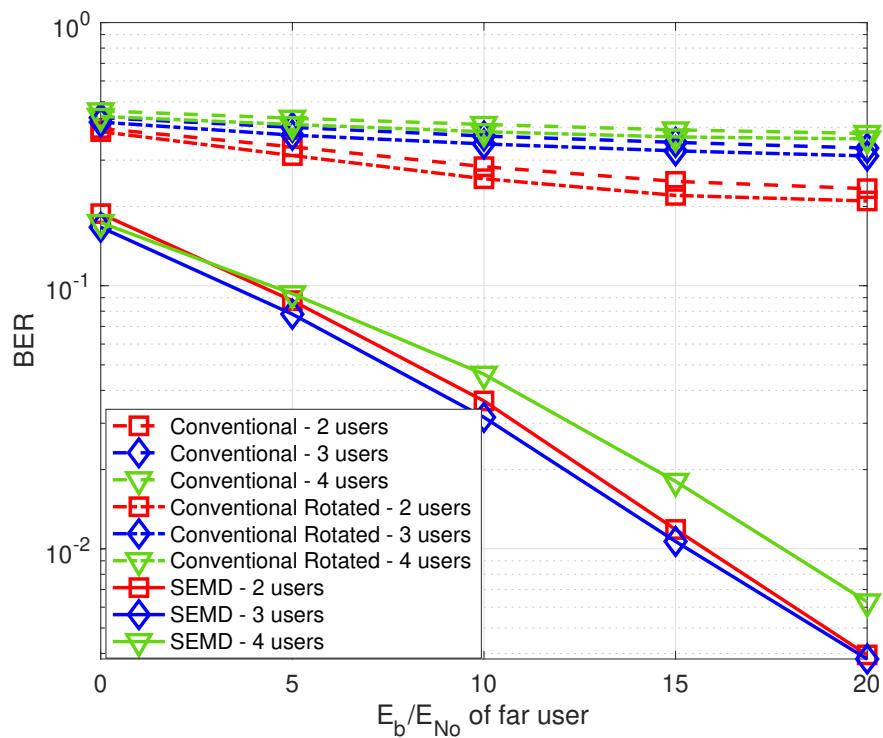
In Fig. 8.4.6, 8.4.7, we can see that both the AE-based SEMD and MEMD frameworks' average BER reduces with SNR due to similar reasons as Remark 31. Also, we can decode the signal without utilizing CSI knowledge. This can be explained by Remark 34 below.

Remark 34 *Firstly, we minimize the binary CE by maximizing the bit-wise MI of all the multiple users together in the downlink network. From a modulation perspective, the constellation design takes place in $2n$ -dimensional space, possibly leading to the maximization of the minimum euclidean distance between the constellation points. From a channel-coding perspective, AE performs the d-BCM designs to maximize the minimum Hamming distance between the codewords. Thus, the proposed AE frameworks can learn the codeword design in higher dimensional space, such that no CSI knowledge is required to decode any of the users.*

As expected, with the increasing number of users, the average BER performance gains of the proposed AE-based frameworks reduce the packing space for a larger number of codewords because of the multiple users in the network utilizing the same higher-dimensional space. Further, the AE-based SEMD framework performs better than the AE-based MEMD framework, especially when the number of multiplexed downlink users increases, even though the SEMD framework has J times less complexity than the MEMD framework. This can be explained by the Remark 35 below.



(a) SEMD - Δ SNR = 0 dB



(b) SEMD - Δ SNR = 6 dB

Figure 8.4.6: Performance evaluation for AE-based SEMD framework performing d-BCM design with $(n, k_j) = (7, 4)$.

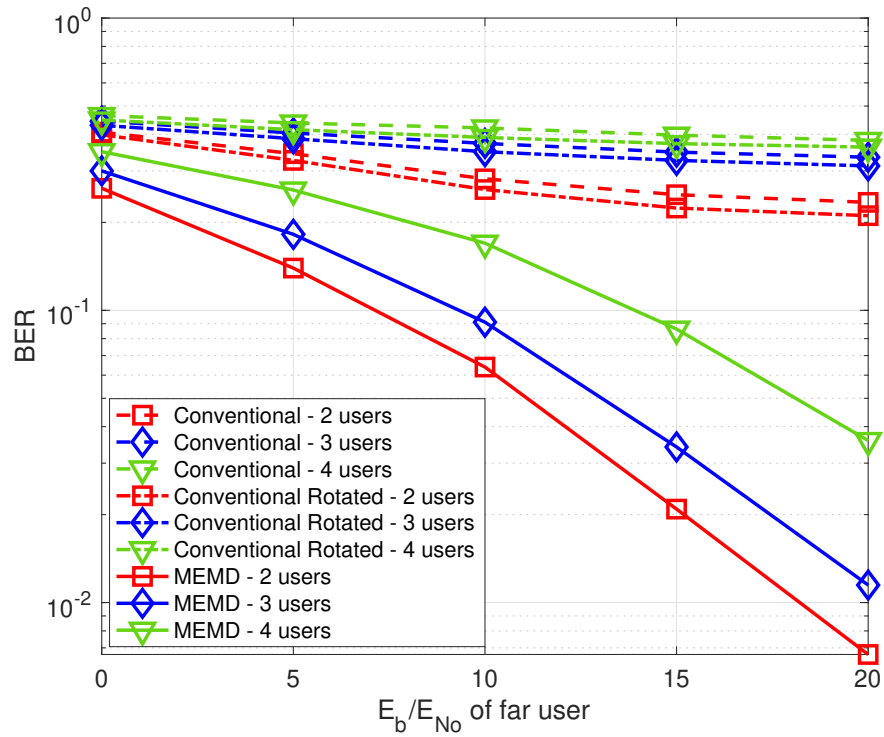
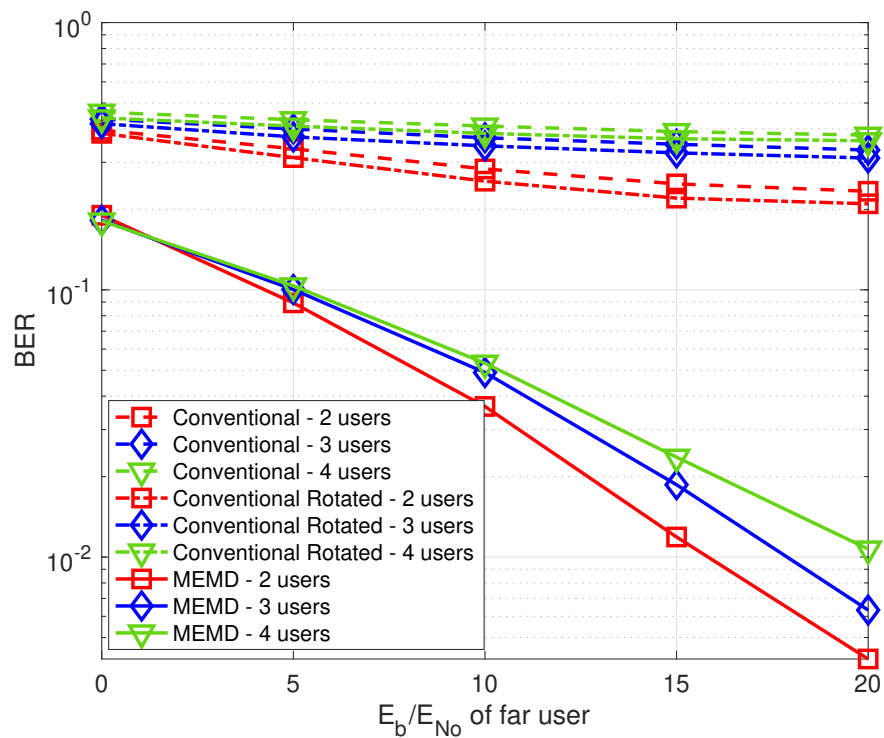

 (a) MEMD - Δ SNR = 0 dB

 (b) MEMD - Δ SNR = 6 dB

 Figure 8.4.7: Performance evaluation for AE-based MEMD framework performing d-BCM design with $(n, k_j) = (7, 4)$.

Table 8.4.3: Performance gains of the proposed AE-based SEMD and MEMD frameworks performing BCM designs over the PD-NOMA with conventional and rotated modulation designs with $(7, 4)$ Hamming code, for $E_b/N_0 = 5$ dB at the farthest user.

No. of downlink users	SEMD				MEMD			
	$\Delta\text{SNR} = 0$ dB		$\Delta\text{SNR} = 6$ dB		$\Delta\text{SNR} = 0$ dB		$\Delta\text{SNR} = 6$ dB	
	Conv.	Rotated	Conv.	Rotated	Conv.	Rotated	Conv.	Rotated
2	4.5 dB	3 dB	3.5 dB	2 dB	4 dB	2 dB	3 dB	1.5 dB
3	8.2 dB	5 dB	6.5 dB	3 dB	7.5 dB	4 dB	3 dB	0 dB
4	11 dB	8 dB	7 dB	4 dB	9 dB	6 dB	4.2 dB	1.8 dB

Remark 35 *Interestingly the lower complexity of the SEMD framework is serving as an advantage, this is because the SEMD framework takes a concatenated bit information of all the users together and designs a single NN-based mapping function that performs joint block channel-coding, modulation design, and signal superposition, as seen in (8.4). In contrast, the MEMD framework is performing the joint block channel-coding and modulation design. However, it performs the signal superposition separately, as seen in (8.7). This shows that with multiple NN-based encoders at the BS, each of the encoders generates its symbols optimally, but the lack of proper superposition of these designed symbols leads to the lower performance gains compared to jointly superposing the symbols using a single NN-based encoder in the SEMD framework.*

AE-based BCM Design

In Fig. 8.4.8, 8.4.9 we can see that both the PD-NOMA and proposed AE-based frameworks can decode the signals because of the presence of the CSI knowledge. Also, the average BER is improved with the SNR due to similar reasons as Remark 31. Moreover, both the proposed AE-based SEMD and MEMD frameworks outperform the conventional NOMA for varying SNR gaps between the users due to similar reasons as Remark 34.

We compare the BER performance gains achieved by the proposed AE-based BCM designs using the SEMD and MEMD frameworks over the conventional NOMA in Fig. 8.4.8, 8.4.9, while summarizing the BER gains in Table 8.4.3 for $E_b/N_0 = 5$ dB at the farthest user. As the number of multiplexed users increases, the BER performance gains achieved by the proposed AE-based BCM design also increase due to similar reasons as Remark 33. Moreover, the proposed AE-based BCM design is obtaining higher gains as the SNR gap between the users reduces, explained similar to Remark 31.

In Fig. 8.4.9a, 8.4.9b, we can see that the BER performance gains of the AE-based MEMD framework performing BCM design reduces with the increasing number of users and becomes the same or worse than NOMA employing rotated modulations with $(7, 4)$ Hamming code. This is due to similar reasons as in Remark 35.

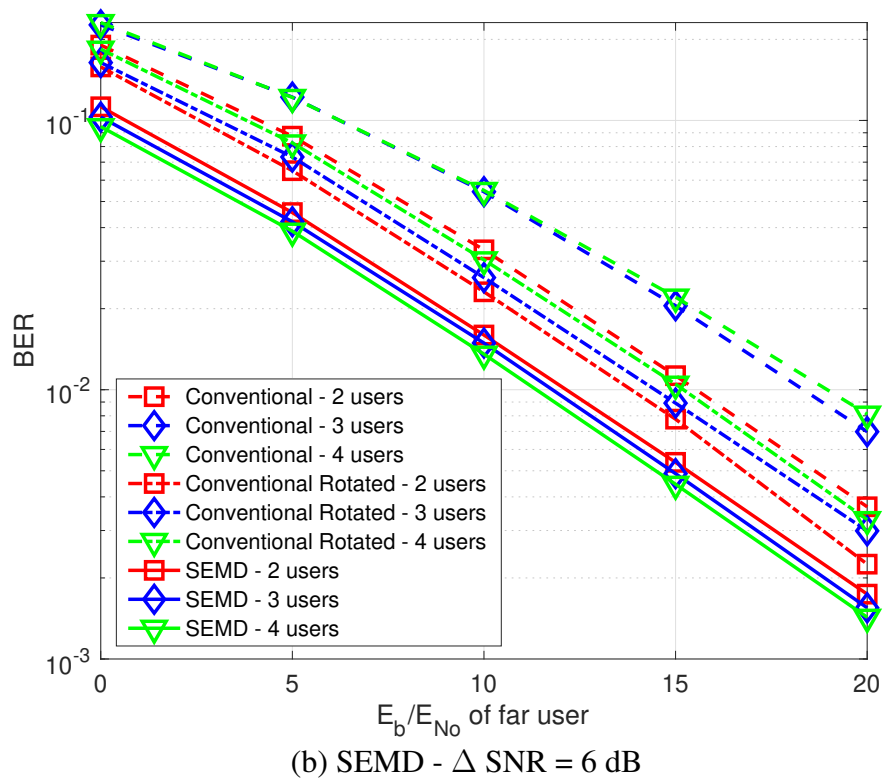
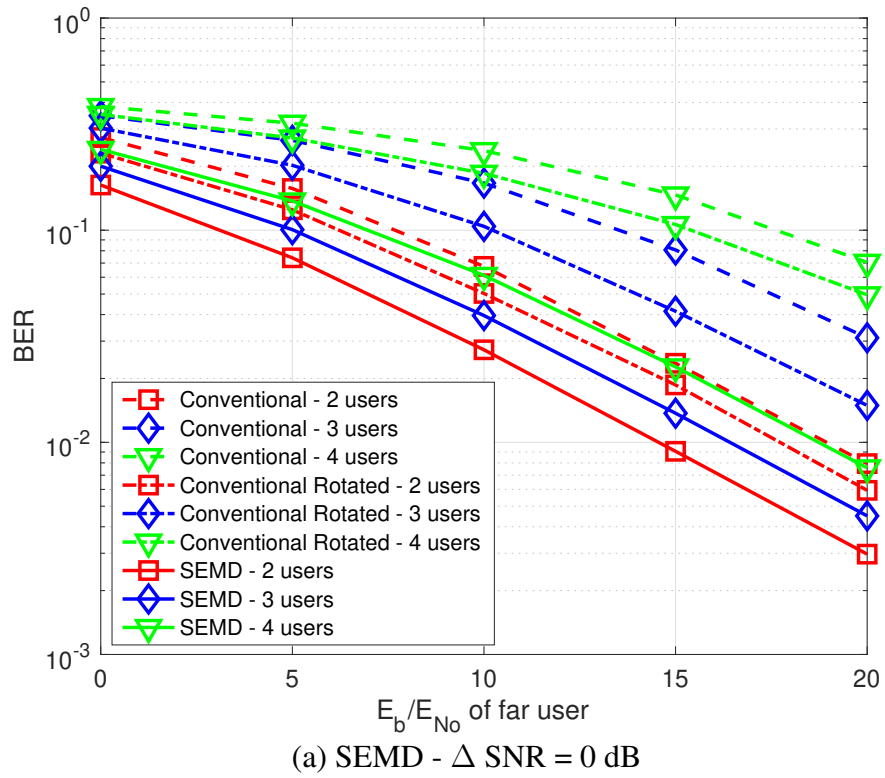


Figure 8.4.8: Performance evaluation for AE-based SEMD framework performing BCM design with $(n, k_j) = (7, 4)$.

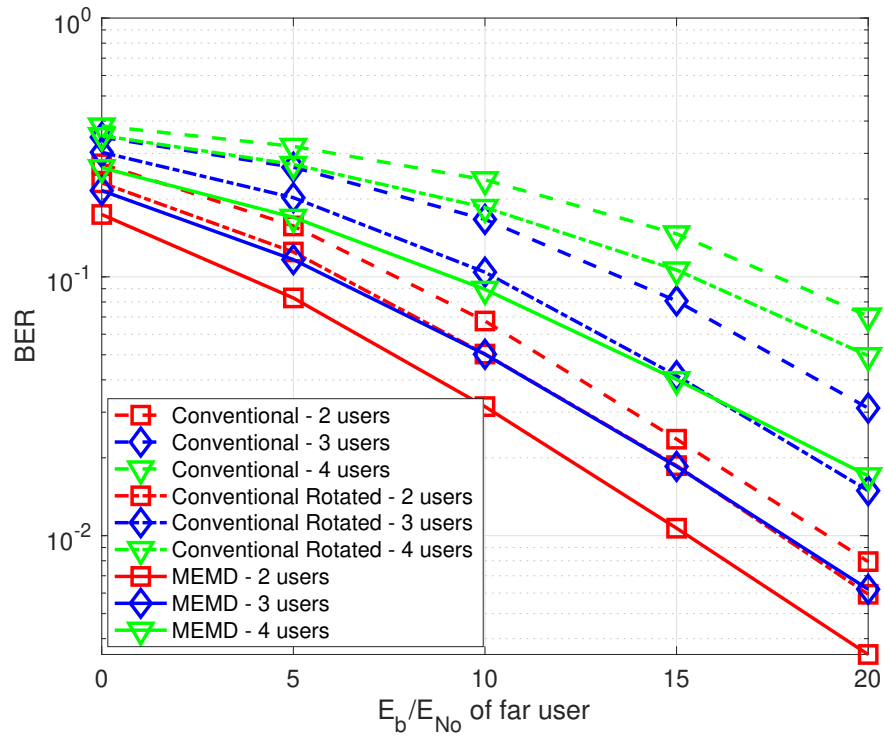
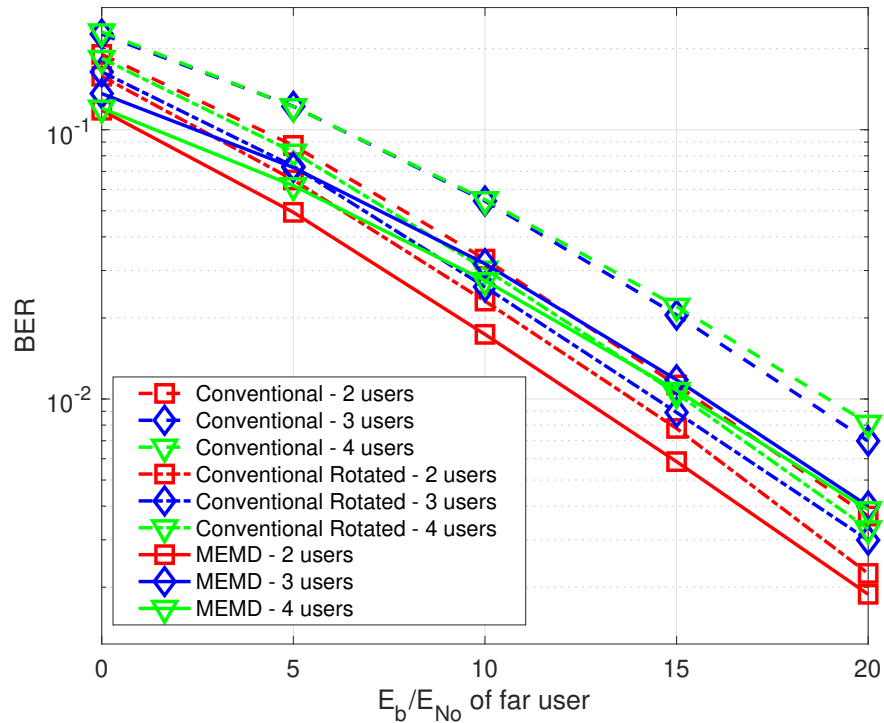

 (a) MEMD - Δ SNR = 0 dB

 (b) MEMD - Δ SNR = 6 dB

 Figure 8.4.9: Performance evaluation for AE-based MEMD framework performing BCM design with $(n, k_j) = (7, 4)$.

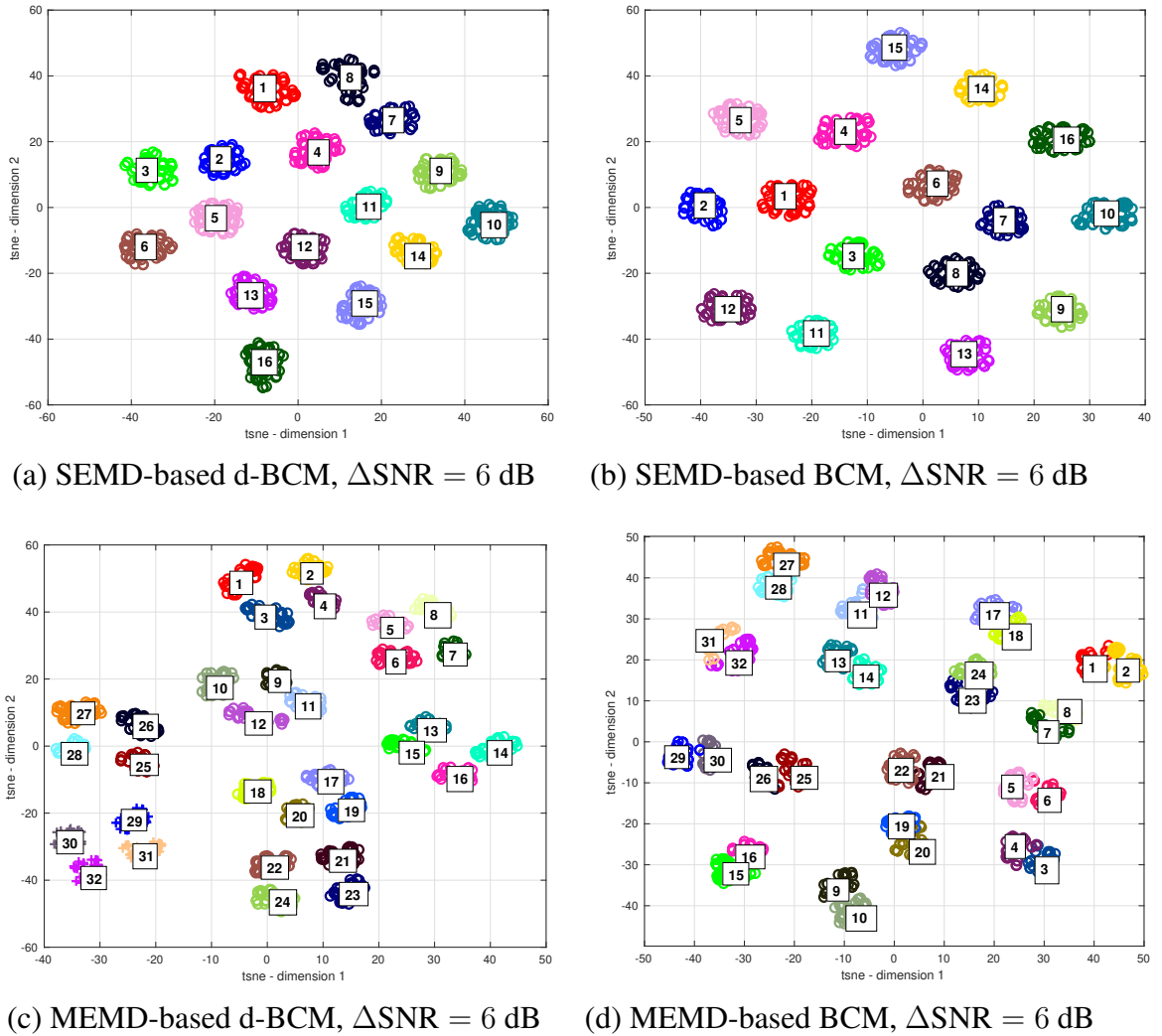


Figure 8.4.10: Bit mapping for AE-based SEMD and MEMD frameworks performing BCM and d-BCM designs with $(n, k_j) = (7, 4)$.

Table 8.4.4: Bit-mapping table for the codewords designed using AE-based SEMD framework.

<i>class</i>	<i>Bit-5</i>	<i>Bit-6</i>	<i>Bit-7</i>	<i>Bit-8</i>
1	0	0	1	0
2	0	1	1	0
3	0	1	0	0
4	0	0	1	1
5	0	1	1	1
6	0	1	0	1
7	0	0	0	1
8	0	0	0	0
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	1	1	0
13	1	1	0	0
14	1	0	1	1
15	1	1	1	1
16	1	1	0	1

Insights for the Codewords Designed by the NN-based Encoder of the AE Frameworks

We now bring insights to the codewords designed by the AE-based SEMD and MEMD frameworks performing BCM and d-BCM designs. Once the AE framework is trained, the NN encoder of the AE-based SEMD and MEMD frameworks becomes deterministic. Thus, if we input any k_j bits of the J users to the NN encoder of the trained AE, we obtain the same n complex baseband symbols as output every time, representing a superposed codeword for the k_j input bits of the J users. Now, we can obtain all the possible codewords from the NN encoders of the SEMD and MEMD framework using all the possible combinations of k_j input bits of the J users. We obtain $2^{\sum_{j=1}^J k_j}$ codewords in the $2n$ -dimensional space.

As our first step, we utilize the t-SNE algorithm, as detailed in Definition 15, to visualize this $2n$ -dimensional data in 2-dimensional space. Secondly, we utilize the k -medoids [121] to cluster the t-SNE results into k subsets or clusters so that the subsets minimize the sum of distances between the t-SNE results and a center of the t-SNE result's cluster, where k becomes a hyper-parameter². We decide this k to be 16 for the SEMD framework and 32 for MEMD after the visualization of the t-SNE results. Lastly, we map the bits transmitted with each cluster to obtain the codeword mapping.

In Fig. 8.4.10, we show the t-SNE plots, wherein we analyze the codewords' mapping obtained for $(n, k_j) = (7, 4)$ for the AE-based d-BCM and BCM designs, while each of the marked cluster of codewords (also called it's class) is described in Table 8.4.5. In this

²Please note k indicates the number of clusters obtained using the k -medoids method, while k_j represents the input bits of the j^{th} user.

Table 8.4.5: Bit mapping table for the codewords designed using the AE-based MEMD Framework, where X represents the *Bit-1* (for the d-BCM design) and *Bit-3* (for the BCM design).

$class$	X	$Bit-5$	$Bit-6$	$Bit-7$	$Bit-8$
1	1	1	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	1	0	0	0
5	0	0	0	0	0
6	1	0	0	0	0
7	1	0	0	0	1
8	0	0	0	0	1
9	0	1	1	0	0
10	1	1	1	0	0
11	0	1	1	0	1
12	1	1	1	0	1
13	0	0	1	0	1
14	1	0	1	0	1
15	0	0	1	0	0
16	1	0	1	0	0

$class$	X	$Bit-5$	$Bit-6$	$Bit-7$	$Bit-8$
17	0	1	0	1	1
18	1	1	0	1	1
19	0	1	0	1	0
20	1	1	0	1	0
21	0	0	0	1	0
22	1	0	0	1	0
23	0	0	0	1	1
24	1	0	0	1	1
25	0	0	1	1	0
26	1	1	1	1	0
27	1	1	1	1	1
28	0	1	1	1	1
29	0	1	1	1	0
30	1	0	1	1	0
31	0	0	1	1	1
32	1	0	1	1	1

subsection, we consider a two-user scenario. Thus we are transmitting 8 bits in total. The bits 1 – 4 belong to the first user (with lower receiving SNR), and bits 5 – 8 belongs to the second user (with ΔSNR dB higher receiving SNR).

For a higher SNR gap between the users, i.e., $\Delta\text{SNR} = 6$ dB in Fig. 8.4.10, we can see that the clusters of data in t-SNE representation are sparsely packed. It can be seen that AE-based joint coding and modulation is making 16 clusters for SEMD and 16 main clusters (with two smaller clusters within) for MEMD, each representing the bits of the second user. This signifies that since the receiving SNR of the second user is high, the AE-based framework is designing a policy to easily differentiate between the second user’s symbols. Moreover, for SEMD, the bits of the first user (i.e., bits 1 – 4) lie within each of the 16 clusters. In contrast, MEMD can make 2 smaller clusters within each of the 16 clusters, based on 1st bit (in case of d-BCM design) and 3rd bit (in case of BCM design) while the other bits of the first user lies within each cluster. Moreover, the AE-based method can design a policy of placing the codewords even when the receiving SNR is low and when $\Delta\text{SNR} = 0$ dB. However, the clusters are very randomly packed, and no direct mapping is obtained. Thereby making its analysis difficult, left for our future works.

Further, for BCM design, the performance gains for the proposed AE frameworks over the conventional NOMA *reduces* with increasing SNR gaps as can be seen in Fig. 8.4.8, 8.4.9 because of reasons elaborated in Remark 31. However for d-BCM design, the performance gains for the proposed AE frameworks over conventional NOMA *increases* when the SNR gap between the users increases from $\Delta\text{SNR} = 0$ dB to $\Delta\text{SNR} = 6$ dB as depicted in Fig. 8.4.6, 8.4.7. This is because the conventional NOMA is unable to decode

the signals of the users, whereas the AE-based framework is able to design the codewords in $2n$ -dimension as can be seen in Fig. 8.4.10, where the second user is properly separated for $\Delta\text{SNR} = 6$ dB.

Moreover, in Table 8.4.4, 8.4.5, we can see that the codewords (clusters or classes) obtained in t-SNE visualization are arranged in a Gray-coded format. This shows that the AE-based framework is placing the clusters with minimum Hamming distance close to each other to improve the BER performance of the multi-user downlink network.

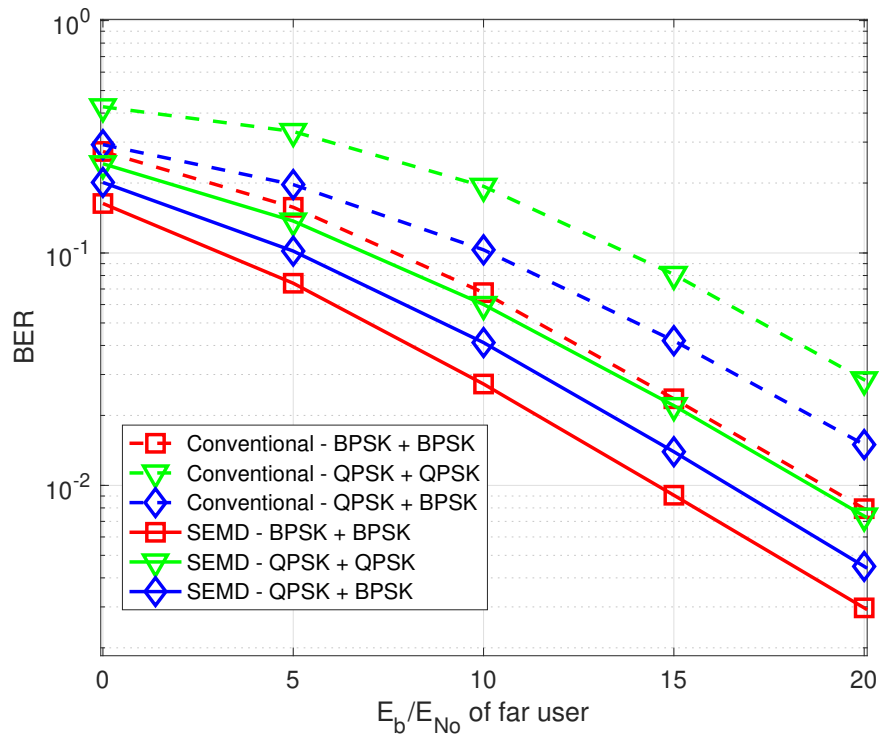
8.4.3 Higher transmission rates

In this subsection, we evaluate the proposed AE-based SEMD and MEMD frameworks performing BCM designs for higher transmission rates, under i.i.d. RBF channels. We consider a two downlink users scenario with near user operating at higher or similar modulation as far user. Also, the PAF (α_j^P) is found to be optimal as 0.2 for the conventional NOMA, where we utilize BPSK + BPSK ($k_1 = k_2 = 4$), QPSK + BPSK ($k_1 = 8, k_2 = 4$), QPSK + QPSK ($k_1 = 8, k_2 = 8$) and 16-QAM + BPSK ($k_1 = 16, k_2 = 4$) modulation at the near and far users, with $(7, 4)$ Hamming codes. Corresponding to the same $(n = 7, k_j)$ we perform BCM design using the proposed AE-based SEMD and MEMD frameworks.

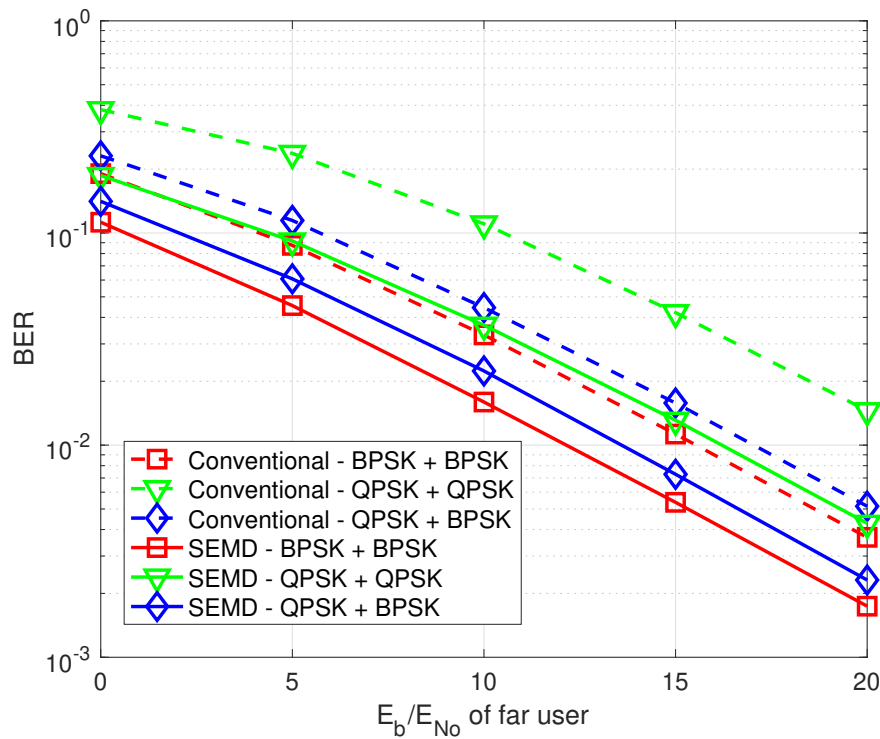
In Fig. 8.4.11, 8.4.12 we can see that the average BER reduces with the receiving SNR. Moreover, we obtain high BER performance gains for the BCM design using the proposed AE-based SEMD and MEMD frameworks, even for varying SNR gaps, over the conventional NOMA. This can be explained by Remark 31 and Remark 34.

Further, we summarize the performance gains achieved by the proposed models over the conventional methods in Fig. 8.4.11, 8.4.12 for $E_b/N_0 = 10$ dB at the far user in Table 8.4.6. Clearly, as the transmission rates for the user increase, the BER performance gains of the BCM design by the proposed SEMD and MEMD frameworks over the conventional NOMA increase. As the transmission rates increase, the number of codewords to be packed in the $2n$ -dimensional space (for proposed AE) and 2-dimensional space (for conventional NOMA) also increases. Further reasoning can be understood from Remark 34. Also, we can see that as the SNR gap between the users' increases, the BER performance gains of the proposed AE-based frameworks reduce due to similar reasons as Remark 31.

Moreover, in Fig. 8.4.13 we consider 16QAM + BPSK ($k_1 = 16, k_2 = 4$) scenario, and we can see that BCM design by the proposed SEMD framework outperforms the conventional NOMA, whereas the BCM design by the proposed MEMD framework is unable to outperform the conventional NOMA. This shows that the MEMD framework has a problem in determining the optimal location for the codewords in the $2n$ -dimensional space, even with J times higher complexity at the NN-based encoder of the BS. However, the BCM design by the SEMD framework outperforms the conventional NOMA



(a) SEMD - Δ SNR = 0 dB



(b) SEMD - Δ SNR = 6 dB

Figure 8.4.11: Autoencoder-based end-to-end learning for higher rates, $n = 7$

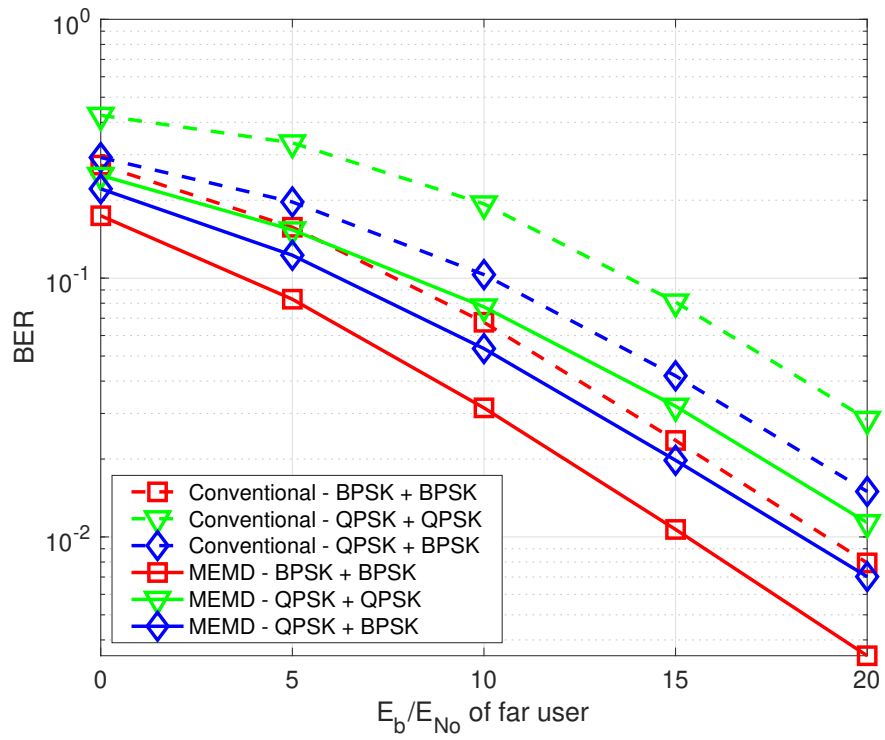
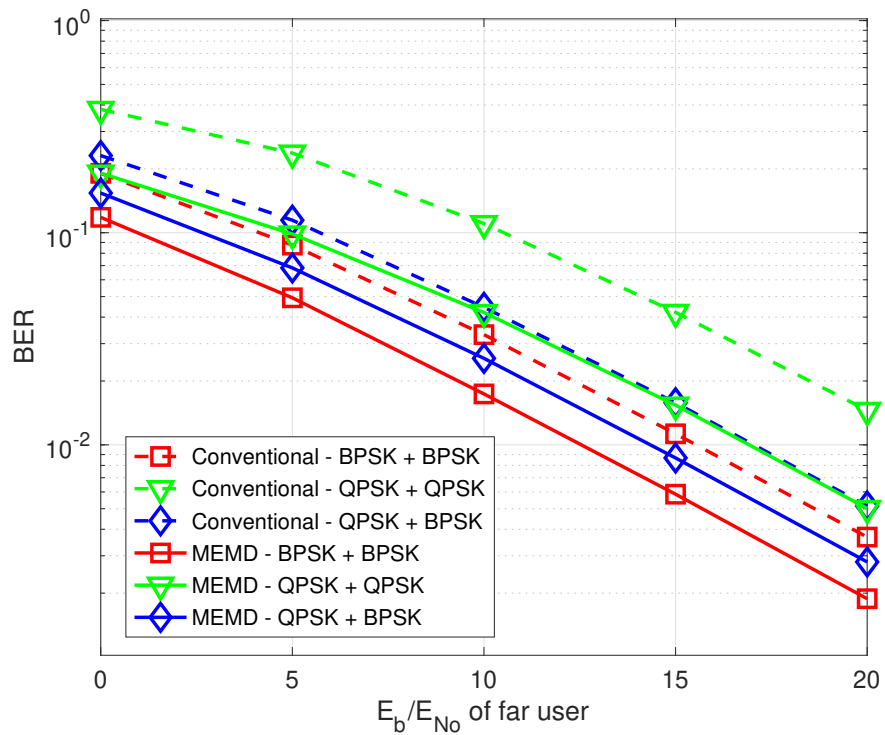

 (a) MEMD - Δ SNR = 0 dB

 (b) MEMD - Δ SNR = 6 dB

 Figure 8.4.12: Autoencoder-based end-to-end learning for higher modulation order, $n = 7$

Table 8.4.6: Performance gains of the proposed over the conventional models for $n = 7$ at received SNR of 10 dB at the far user for higher transmission rates.

Modulation (Near + Far user)	SEMD		MEMD	
	$\Delta\text{SNR} = 0$ dB	$\Delta\text{SNR} = 6$ dB	$\Delta\text{SNR} = 0$ dB	$\Delta\text{SNR} = 6$ dB
BPSK + BPSK	4.5 dB	3.5 dB	3.5 dB	3 dB
QPSK + BPSK	5 dB	3.5 dB	4 dB	2.8 dB
QPSK + QPSK	6.5 dB	6 dB	5.2 dB	5 dB

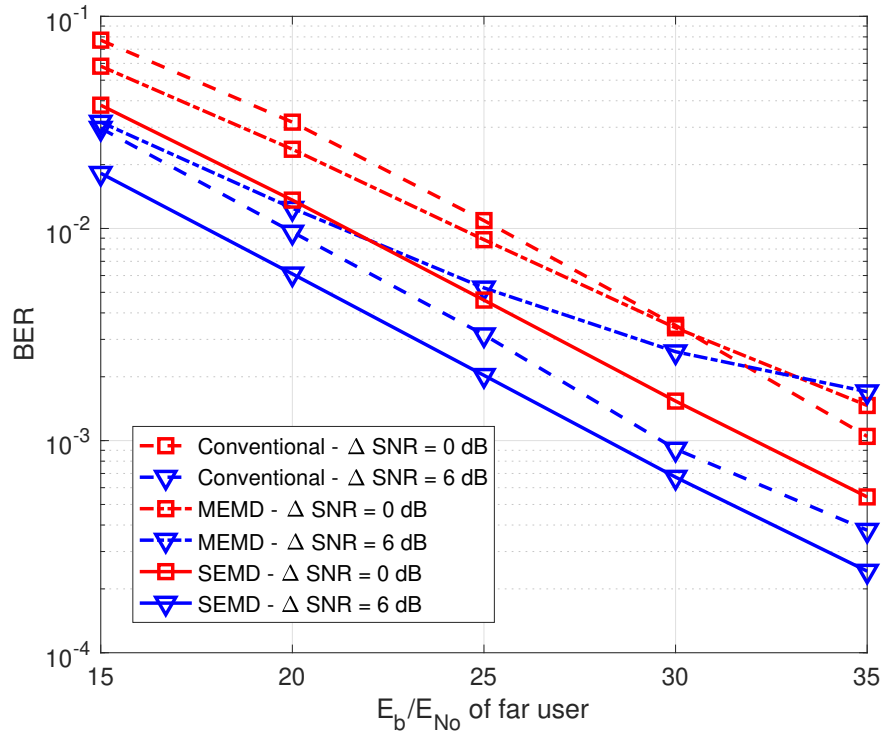


Figure 8.4.13: 16 QAM + BPSK

by 4 dB for users at same SNR ($\Delta\text{SNR} = 0$ dB) and 2 dB for users at different SNRs ($\Delta\text{SNR} = 6$ dB) at $E_b/N_0 = 25$ dB for the far user, respectively. Thus, we can say that the proposed SEMD framework outperforms the proposed MEMD framework when the number of users increases, transmission rates increases, and channel reuse increases due to similar reasons as Remark 35.

8.5 Conclusion

In this chapter, we proposed two end-to-end learning-based bit-wise Fixed AE frameworks for multi-user downlink networks - single encoder multiple decoders (SEMD) and multiple encoders multiple decoders (MEMD), under Rayleigh fading channels. We optimize the AE frameworks by maximizing the total bit-wise MI of all the multiple users in the downlink network. Furthermore, for the AE-based modulation design, we remove

the SIC at the strong users and the requirement of the PAF at the BS. Thereby removing the need to feedback channel gains to the BS, PAF optimization, error propagation due to multiple SIC, and the need to send the SIC ordering and PAF information to the multiple downlink users. Moreover, since the proposed AE frameworks remove the SIC thus, we can decode the superposed signal of two users on the same SNR, removing the necessity of users to have a minimum of 3 – 8 dB of SNR gap for the successful implementation of SIC. Further, we analyzed the minimum squared Euclidean distances for the constellation obtained by the proposed AE frameworks to infer the reasons behind the BER performance gains and user fairness achieved by the proposed QoS constraint in the AE frameworks. Furthermore, as the number of users increases, we show that the BER performance gains of the proposed AE frameworks increase over the conventional NOMA. Apart from achieving these benefits, for the AE-based BCM and d-BCM design, we show that we can decode all the multiplexed users' signals using the proposed AE frameworks even without the CSI knowledge at the receivers, which remains mandatory to implement SIC in conventional NOMA. Furthermore, we utilized t-SNE and k-medoids to understand the bit-mapping of the codewords designed in the $2n$ -dimensional space. We also show that our proposed AE frameworks perform better with increasing transmission rates. Lastly, we show that the AE-based SEMD framework outperforms the AE-based MEMD framework when the number of users increases, transmission rates increases, and channel reuse increases, even with J times lower complexity at the BS.

Chapter 9

AE-based Feature Compression and Machine Learning (ML)-based Mm-Wave PL Prediction

9.1 Introduction

The 5th generation of mobile networks (5G) has adopted a much broader spectrum at higher frequency bands, such as mm-wave bands, that promises very high data rates. To unleash the full potential of mm-wave communications, highly accurate channel modeling and path loss (PL) prediction are essential in foretelling cell coverage, planning deployment of the base stations (BSs), and optimizing network performance [122]. However, high bands come with the challenge of higher free-space, scattering, and diffraction losses from the propagation environment. For example, in a typical urban street, buildings and street clutter like scaffolding, vehicles, and tree canopies can significantly impact PL compared to lower frequency bands (wavelength of tens of cm). Although accurate PL estimation by employing fast and straightforward models is pivotal in network planning and optimization, they are yet to be fully understood in mm-wave frequencies at various propagation environments.

Numerous PL prediction models have been established in the literature, which can be classified into three major categories: statistical [123, 124, 125, 126, 127, 128], deterministic [129], [130], and learning-based models [131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150]. Statistical models such as [123] provide a computationally efficient method by fitting particular equations to measurements obtained in different propagation environments [123, 124, 125, 126, 127, 128]. The most widely adopted heuristic channel models, referred to as slope-intercept model hereafter, apply a linear fit to the measured PL data against the logarithm of the Euclidean distance between the transmitter and the receiver. Deterministic models such as

ray-tracing, on the other hand, are established based on the principles of physics using complex set of user-specified simulation parameter settings, and its performance relies on an accurate description of the propagation environment [129, 130]. Field measurements at 28 GHz [128] have shown that ignoring street clutter leads to 10 dB underestimation of PL in urban street canyons. However, the level of detail in the environment description, including both shape and material property, required at mm-wave bands are challenging to characterize using conventional statistical or deterministic modeling. Therefore, machine learning (ML) based techniques have appeared as a promising alternative.

9.1.1 Previous Works

PL prediction can be considered as a regression problem in ML, where the features extracted from the propagation environment become its input and PL as a continuous variable output. We summarize some of the ML-based approaches for propagation environment modeling and PL prediction [131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150] in Table 9.1.1, highlighting the propagation environment, frequency, key features, training and testing procedures, PL data source, and ML tools such as artificial neural networks (ANNs), random forest (RF), convolutional neural network (CNNs), autoencoder (AE), and support vector regression (SVR). These works showcased the capability of ML-based methods and their potential in improving PL prediction accuracy. A more comprehensive review on ML-based PL prediction can be found in [151].

Many of the ML-based approaches [131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 149] focus on prediction for nearby links (i.e., interpolation). For studies ([144, 145, 146, 147, 148, 150]) that do predict PL for new streets/areas (i.e., extrapolation), the influence of street clutter such as trees and street furniture on PL is either minimal or non-existent. Besides, in most previous works, complex ML models are adopted as a black-box, making it hard to interpret the connection between features and PL prediction. These complex ML models also make PL prediction vulnerable to overfitting since the training data size from measurements is usually too small compared to adjustable parameters in ML models.

Compared to studies such as [131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148] that are dedicated to the sub-6 GHz bands, ML-based PL prediction for mm-wave bands requires a much finer level of details in the environment description as scattering by small objects (tenths of wavelength) and material absorption loss are more significant for mm-wave signals. For example, about 30 dB street-by-street variation in median PL has been observed from field measurements in urban street canyons at 28 GHz [128]. To capture street specific features, 2D-collapsed images of 3D-building data have been used in [149] for PL prediction using CNN-based model.

Table 9.1.1: Comparison of learning based path loss prediction [131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150].

<i>Ref</i>	<i>Environment</i>	<i>Frequency</i>	<i>Key Features</i>	<i>ML tools</i>	<i>Train/test</i>	<i>Data Source</i>
[131]	Indoor	1.89 GHz	Tx Position and gain, Tx height, distance	ANN	Split dataset	Measurements
[132]	Rural	0.881 GHz	Tx height, TLA, land usage	ANN	Split dataset	Measurements
[133]	Desert like area	1.8 GHz	Terrain profile	ANN, CNN	Split dataset	Measurements
[134]	Railway	0.930 GHz	Viaducts, cuttings, plains	ANN	Split dataset	Measurements
[135]	Urban	0.9, 1.8 GHz	Building height, LOS path	ANN, RF	Split dataset	Simulated
[136]	Urban	0.9 GHz	2D satellite images	VGG-16	Split dataset	Simulated
[137]	Urban	2.6 GHz	Tx gain, position and height, distance	ANN	Split dataset	Measurements
[138]	Urban	2.1 GHz	3D point cloud	ANN	Split dataset	Measurements
[139]	Urban	–	Propagation loss, distance	ANN	Split dataset	Measurements
[140]	Urban	3 – 6 GHz	Distance, frequency	ANN	Split dataset	Measurements
[141]	Urban	5.8 GHz	Antenna separation, distance, frequency	ANN, RF, SVR	Split dataset	Measurements
[142]	Urban, Suburban, Rural	2.4 GHz	Satellite images used for image segmentation	AE	Split dataset	Measurements
[143]	Alpine (Dense Trees)	2.4 GHz	Distance, vegetation variability, terrain, canopy coverage	ANN, RF, kNN	Split dataset	Measurements
[144]	Urban, Rural, Suburban	1.8 GHz	Longitude, latitude, altitude, clutter height, elevation	ANN	route-wise split	Measurements
[145]	Desert like area	1.8 GHz	Terrain profile	RF AE	Spatially disjoint split	Measurements
[146]	Public square	high frequency	Building height, distance, LOS/NLOS	CNN	site-wise split	Simulated
[147]	Urban	2.6 GHz	Satellite images	CNN	site-wise split	Measurements
[148]	Urban	5.9 GHz	2D city map, street map, location of cars	CNN	map-wise split	Simulated
[149]	Urban	28 GHz	3D-buildings	CNN	Split dataset	Simulated
[150]	Urban	28 GHz	Building height, terrain height, tree foliage height, and LOS information	CNN-based U-Net	Split dataset & city-wise split	Simulated

In [150] building height, terrain height, tree foliage height, and LOS indication have been used as input images to the CNN-based U-Net image segmentation model. Both models in [149] and [150] are trained by minimizing the difference between predicted PL and ray-tracing PL. However, simulated data from ray-tracing has been found to be unreliable (over 10 dB underestimation of PL) in urban street canyons at 28 GHz [128] when street clutter is ignored¹. Therefore, high quality PL training data collected from field measurements (which is time and resource consuming) is crucial to establish reliable PL prediction models.

9.1.2 Our Contributions

We address three key challenges in ML-based PL prediction for mm-wave bands: 1) *reliability* due to no/insufficient measurement data; 2) *generalizability* due to weak/no capability of extrapolation; 3) *interpretability* due to complex ML models and high dimensional features.

Reliability

We utilize a large-scale dataset from 28 GHz field measurements [128] in urban street canyons, consisting of 1028 continuous-wave links from 13 streets in Manhattan from multiple roof top sites co-located with commercial BS. The street clutter information such as tree canopies and lampposts is obtained from the open-source LiDAR point cloud dataset [152], and the building information is obtained from the open-source 3D mesh-grid [153], which includes building height, façade shape, separation between the buildings, roads, elevation information, etc.

Generalizability

We address it from three aspects:

- *Street-by-street training and testing policy*: 13 independent training-testing combinations are created by choosing one street at a time for testing and the rest 12 streets for training. Such policy would test extrapolation of trained models to “never seen” streets.
- *Aggressive street clutter and building feature compression*: For each link, the high-resolution point cloud raw data is compressed to two numbers using heuristic approaches devised from expert-knowledge in wave propagation, and the 3D building

¹Improving the accuracy of ray-tracing for mm-wave in presence of street clutter is a challenging problem by itself.

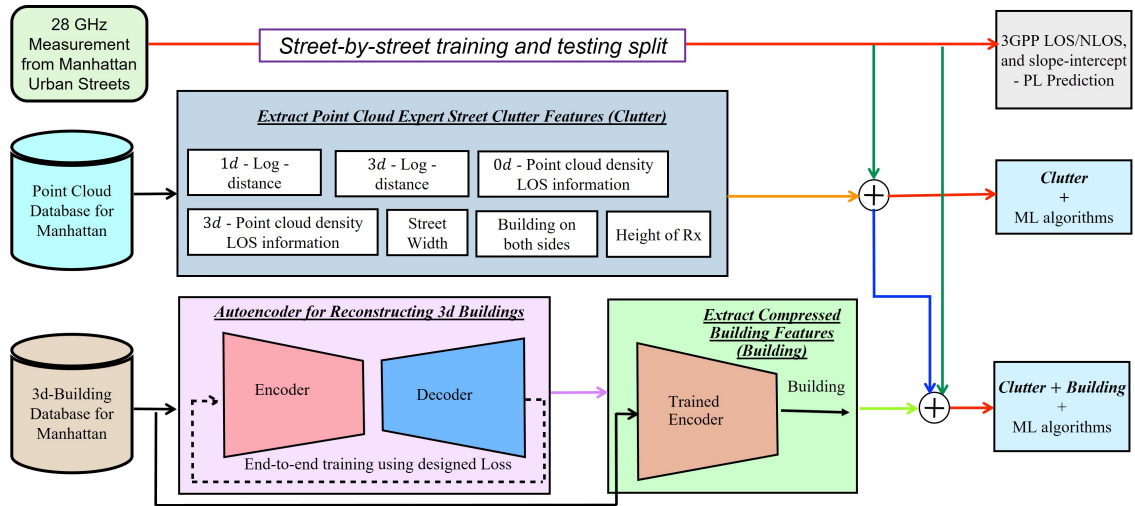


Figure 9.1.1: Methodology adopted in this work.

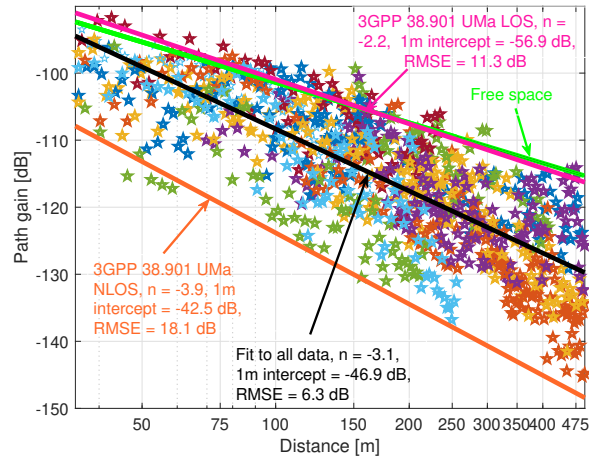
information is compressed to a length-12 vector using CNN-based autoencoders to preserve the spatial characteristics.

- *Reducing adjustable parameters for PL prediction:* Simple ML based regression algorithms such as Lasso, Elastic-net, random forest, and SVR are adopted to mitigate over-fitting.

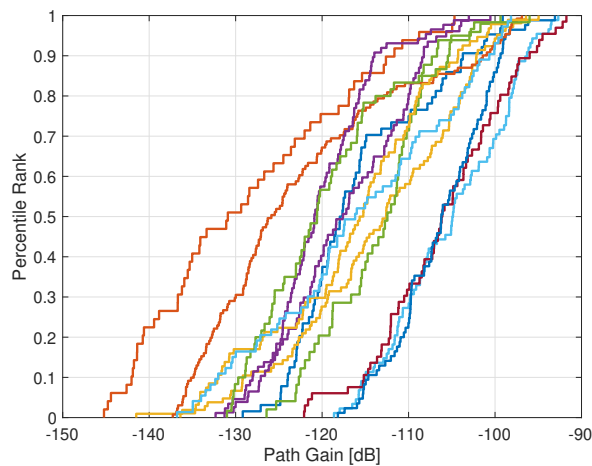
Interpretability

We adopt human-friendly environment features and quantify the significance of each feature in the PL prediction. We define only seven expert-knowledge driven propagation environment features, referred as *Clutter* features hereafter, where each feature has a physical meaning attached. We quantify the importance of each *Clutter* feature by Lasso weight analysis and by comparing the PL prediction accuracy when only one feature is excluded.

To the best of our knowledge, this is the first time both street clutter and building information are used collectively for mm-wave PL prediction using a large-scale real-world propagation measurement at 28 GHz in urban streets. We show that our proposed model achieves root mean square error (RMSE) of 4.8 dB averaged over all 13 streets with 1.1 dB standard deviation that reflects street-by-street variation. By only using the top four most influential features, our model achieves prediction RMSE of 5.5 ± 1.1 dB (mean \pm std). In contrast, the heuristic slope-intercept method and the 3GPP LOS model based prediction have RMSE of 6.5 ± 2.0 dB and 10.6 ± 4.4 dB, respectively. For the first time, we show that the ML assisted PL predictions are more accurate than measurement based slope-intercept model with much smaller street-by-street variation. The methodology adopted in this work is summarized in Fig. 9.1.1.



(a) Measured path gain of 1028 links from 13 streets



(b) Large street-by-street variation

Figure 9.2.1: Manhattan measurement data (different color per street).

9.2 PL Data Collection and Feature Sets Preprocessing

In this section, we present an overview of the PL measurement in Manhattan and preprocessing of the point cloud dataset and the 3D building mesh grid dataset.

9.2.1 28 GHz PL Measurement in Manhattan

The measurement campaign [128] is designed to mimic urban street canyon coverage from roof edge mounted BS (i.e., urban macro) to UEs (1.5 m high) in the center of a sidewalk along the street. No attempt was made to incorporate or eliminate blockage due to street clutter, such as trees, vehicles, pedestrians, scaffolding. The purpose is to resemble coverage of the street in the presence of such obstructions. Measurements were performed from 7 building rooftops with height ranging from 15 to 48 m, covering 13 streets with Tx-Rx distance ranging from 35 to 500 m. In total, 1028 links were measured with over 10 million individual power measurements, which were locally averaged per

link to eliminate small scale fading.

In Fig. 9.2.1a we show the measured links and the slope-intercept fit² to PL versus logarithmic Euclidean distance (d), given by

$$P = A + 10n \log_{10}(d) + \mathcal{N}(0, \sigma^2), \quad (9.1)$$

where $A=46.9$ dB is the 1 m-intercept, $n=3.1$ represents the slope, $\sigma=6.3$ dB is the RMSE between fitted and actual values and $\mathcal{N}(\cdot)$ is the normal distribution representing shadow fading. Comparing our data against standard PL models such as 3GPP UMa LOS and NLOS [154] leads to RMSE of 11.3 dB and 18.1 dB, respectively. Fig. 9.2.1b depicts the distributions of PL for individual streets, with median spanning over a range about 30 dB.

The *presence of street clutter* may be the cause for about 10 dB excess loss compared to the UMa LOS model in Fig. 9.2.1a and the large street-by-street variations in Fig. 9.2.1b. This is due to the short wavelengths of mm-wave signals, approximately 1 cm at 28 GHz, making them more susceptible to intense diffused scattering and poorer rough surface reflection [155]. Further, mm-wave has a tighter first Fresnel zone directly proportional to its wavelength, causing objects as small as tens of centimeters to appear to be substantial in impairing link quality [156]. This motivates us to capture detailed environment features such as street clutter and 3D-building.

9.2.2 Street Clutter Modeling using LiDAR Point Cloud Dataset

Let us consider the measurements done from the same rooftop for two Manhattan streets, the 7th Avenue with a handful of young trees and the W 11th Street with many tall tree canopies, as shown in Fig. 9.2.2a, and Fig. 9.2.2b, respectively. Measured PL and their slope-intercept fits are shown in Fig. 9.2.2c. The distance exponent of the W 11th Street is significantly higher, 8.7, compared to 3.4 on the 7th Avenue, with a 23 dB gap in average PL at 500 m. Therefore, street clutter information, which includes tree canopies, cars, lamppost, etc., plays a crucial role in PL prediction.

To capture the street clutter, we use the *USGS CMGP LiDAR* point cloud repository [152], where each object is described by a set of points on its external surfaces acquired at 1 cm resolution. For each street, we change the origin to the ground location of the Rx position and align X -axis with the street along which the Tx is moving, Y -axis along the width of the street, and Z -axis pointing to the Rx placed at the top of the building. We also utilize a *k-nearest neighbor* based point cloud denoising [157]. The processed point cloud representing the street clutter for the 7th Avenue and the W 11th

²Path gain, instead of path loss, is displayed in Fig. 9.2.1a to emphasize the deteriorating link quality as distance increases.

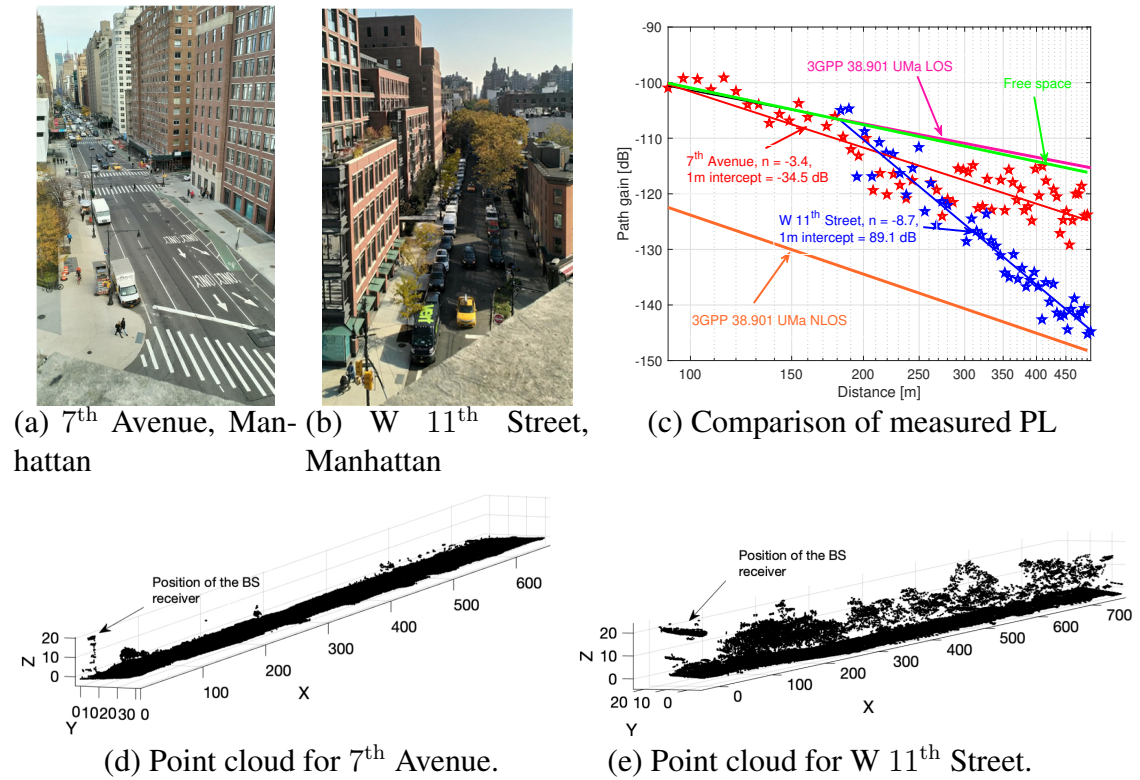


Figure 9.2.2: Two streets covered from the same roof top with diverse street clutter density.

Street are shown in Fig. 9.2.2d, and Fig. 9.2.2e, respectively.

9.2.3 3D-Building Mesh Grid

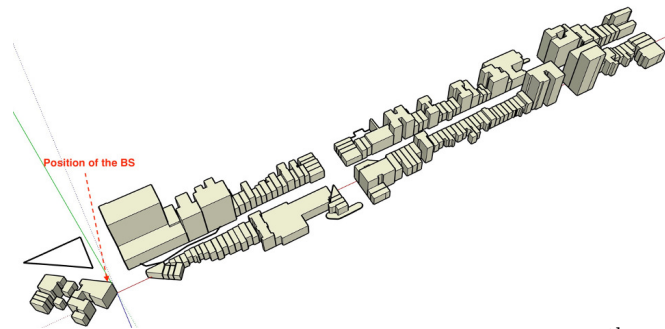
Reflection and scattering from urban buildings can be significant and impose wave-guiding effect on the signals. We extract the 3D building mesh grid from the Cadmapper [153]. The aerial-view of the W 11th Street is shown in Fig. 9.2.3 and the extracted 3D buildings are in Fig. 9.2.4a. We first convert the high dimensional mesh grid into a Euclidean space by assigning each $1 \times 1 \times 1$ m cube a value 1/0 indicating the presence/absence of a mesh grid. We then reposition the origin such that Rx is at $(0, 0, Rx \text{ height})$ and align the X -axis along the street and Y -axis along street width (i.e., align the coordinates with those used for point cloud). To reduce the dimensionality of the dataset while preserving height information, we collapse it along the Z -axis into 2D grids (of 1×1 m) and assign the entry of each grid the height of building at that location (0 if there is no building), as shown in Fig. 9.2.4b for the W 11th Street where the color bar indicates height.

9.3 Street Clutter Feature Extraction from Point Cloud

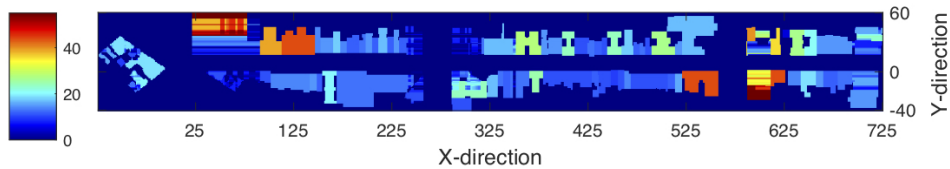
The LiDAR point cloud dataset [152] contains massive amount of data that can't be directly used for training or interpretation. Thus, we focus on expert-knowledge driven



Figure 9.2.3: Aerial-view of W 11th Street.



(a) 3D building (mesh grid) acquired for the W 11th Street.



(b) 2D collapse for the W 11th Street mesh grid buildings.

Figure 9.2.4: Example of 3D building mesh grid data

feature extraction from the point cloud dataset for modeling the street clutter. From expert knowledge, we know that log 3D-distance and 1D-distance between the transmitter and receiver, street width, and the base station height are important parameters, widely used in the 3GPP standard for the PL prediction. Further, due to the tight first Fresnel zone, we know that the maximum signal should come from the LOS direction, while the overall street clutter can help in providing guidance for general characteristics of the clutter in the street. Using this expert knowledge, we obtain the most relevant features from the point cloud data as below.

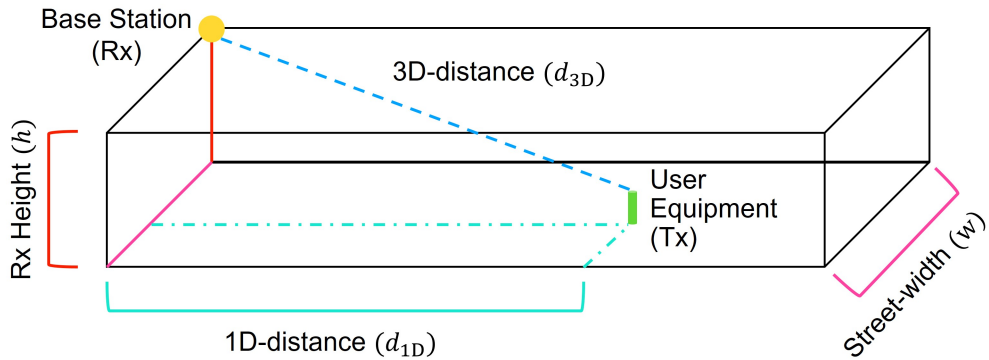


Figure 9.3.1: Illustration of point cloud based Clutter features 1, 2, 5, 7.

9.3.1 Street Clutter Feature Compression

We compress all the LiDAR point cloud information for each link into two numbers using heuristic approaches devised from expert-knowledge in wave propagation, where each number simply equals the count of data points in a 3D volume:

- a string of contiguous $1 \times 1 \times 1$ m cubes from Tx to Rx along the direct path (*3D-info*)
- a street-wide cuboid as high as the BS connecting Rx and **the furthest** Tx (*0D-info*)

The *3D-info* represents the accumulated clutter density along the direct path within the first Fresnel zone (about or smaller than the $1 \text{ m} \times 1 \text{ m}$ cross-section) and thus may be interpreted as a blockage indicator of the direct path for each link. The *0D-info* represents the overall clutter density of the entire street and remains the same for all links from the same street.

9.3.2 Point Cloud based Expert Street Clutter Features (Clutter)

The following seven expert street clutter features are defined for PL prediction:

- *Clutter 1: log-3D distance* - Euclidean distance (log-scale) between the Tx and Rx.
- *Clutter 2: log-1D distance* - Along-the-street distance (log-scale) between the Tx and Rx.
- *Clutter 3, 4: 0D-info and 3D-info* - Clutter density information as defined in Sec. 9.3.1.
- *Clutter 5: Street width* - Spans from 15 to 38 m.
- *Clutter 6: Buildings on both sides* - Indication of guiding effect from street canyon.
- *Clutter 7: Rx height* - Spans from 15 to 48 m.

Table 9.3.1: Point Cloud based Clutter features (3, 5, 6, 7) for the Manhattan measurement campaign.

<i>Street Number</i>	<i>Street Description</i>	<i>#measured links</i>	<i>ID-distance range</i>	<i>OD info</i>	<i>Street width</i>	<i>Building on both sides</i>	<i>Height Rx</i>
1	Car-parking both sides, moderate vegetation	88	32 – 165 m	0.0051	25 m		15 m
2	Two-way road with divider in between having tree-canopies	77	69 – 338 m	0.0271	23 m		35 m
3	Two-way road, hilly-road, car parking, moderate vegetation	94	36 – 492 m	0.0186	28 m		15 m
4	Two-way road, moderate tree-canopies, moving-traffic	105	35 – 453 m	0.0274	36 m		25 m
5	Car-parking both sides, moderate vegetation	85	32 – 163 m	0.0051	25 m		15 m
6	Two-way road, hilly-road, car parking, moderate vegetation	131	34 – 495 m	0.0186	28 m		15 m
7	Street-full of car parking, light vegetation	87	153 – 497 m	0.0170	15 m		20 m
8	No vegetation, moving-traffic, many scaffoldings	64	92 – 490 m	0.0032	30 m		22 m
9	Car-parking both sides, moderate vegetation	66	36 – 235 m	0.0151	27 m		15 m
10	Building on only one side, High vegetation, curved road	73	37 – 217 m	0.0342	27 m	X	40 m
11	Moderate vegetation, car-parking, moving-traffic	49	151 – 411 m	0.0084	38 m		48 m
12	Very high vegetation, car parking both-sides	49	182 – 497 m	0.0504	18 m		22 m
13	Building on only one side, High vegetation, curved road	63	36 – 255 m	0.0418	27 m	X	40 m

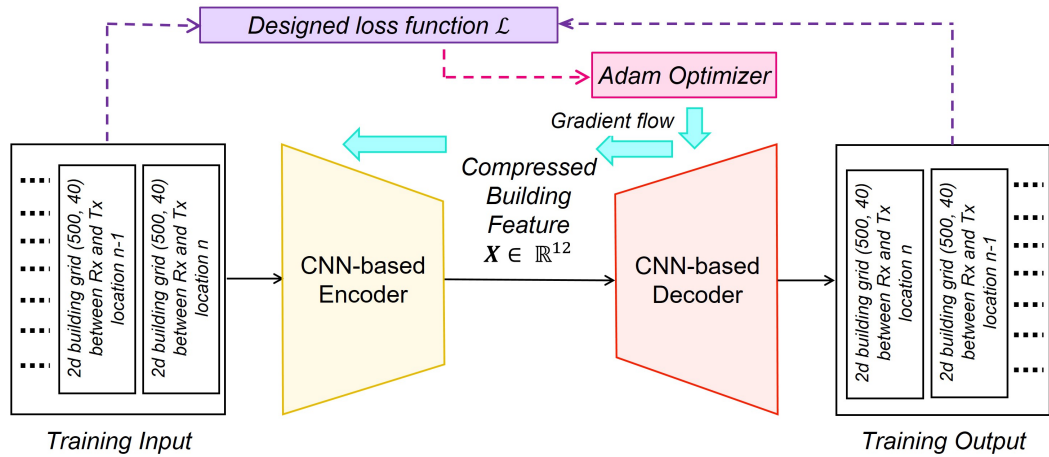


Figure 9.3.2: Proposed CNN-based autoencoder for building feature compression.

Among the defined seven Clutter features, four of them (Clutter 1, 2, 5, 7) have been used in different 3GPP models: 1D distance (d_{1D}), 3D distance (d_{3D}), street width, and the base station or Rx height (h), as shown in Fig. 9.3.1. Such features provide us a direct means to interpret the trained models and compare them against the 3GPP and slope-intercept models. The other three features (Clutter 3, 4, 6) capture street-specific (*0D-info*) and link-specific (*3D-info*) clutter information as well as the street canyon information (building on both sides of the street).

We summarize the defined expert features in Table 9.3.1, wherein we omit *Log-3D distance*, *Log-1D distance*, and *3D clutter information* because they have a separate entry for each link. Besides, we report the range of measurement distances with number of measurement links on each street and the diverse nature of the streets by “street description” (that points to 30 dB street-by-street variation seen in Fig. 9.2.1b), that is coarsely captured by the designed seven Clutter features. We standardize the defined Clutter features before training and testing: for feature f we compute its mean μ and variance σ^2 on the training dataset; then, we rescale the feature in both the training and testing dataset as $\hat{f} = (f - \mu)/\sigma$.

9.4 Autoencoder based Feature Extraction from Building Dataset

Although we have converted the 3D building mesh grid to a 2D matrix representation, as explained in Sec. 9.2.3, the building features are still much richer than the PL data. We further reduce the 2D collapse of buildings to size of (500, 40) by removing buildings beyond the maximum measurement distance 500 m and by including the building façade only (taking grids 20 m from the center of the street on each side). As the Tx-Rx distance increases, the number of 2D grids in between the two also increases. We append zero to

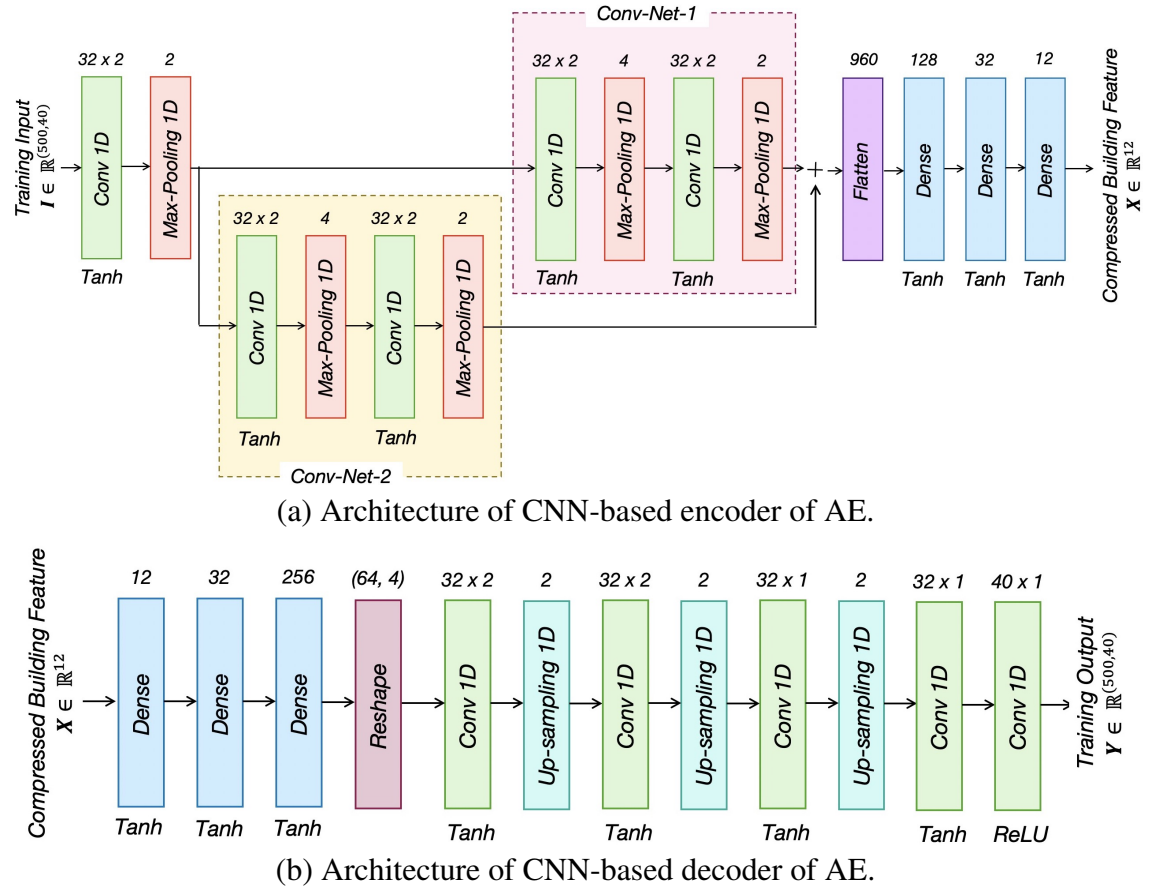


Figure 9.4.1: Neural network architectures for the proposed CNN-based autoencoder.

the 2D matrix representation at shorter distances to preserve the distance dependency and avoid information loss by downsampling.

We then use CNN based AE to compress the extensive building façade data to a few most relevant features for PL prediction. CNN captures the spatial dependencies with the helps of kernels and filters, and AE learns an efficient encoder (for feature compression) and a matching decoder (to reproduce the original input signal) in an unsupervised manner. Therefore, a combination of CNN and AE, as shown in Fig. 9.3.2, can help us to reduce the massive feature dimension of the building data to 12 features while preserving the spatial characteristics.

The encoder in the CNN-based AE, as shown in Fig. 9.4.1a, takes 2D-buildings $I \in \mathbb{R}^{(500,40)}$ as input and reduces it to compressed features $X \in \mathbb{R}^{(12)}$. This is achieved by the convolutional and max-pooling layers that help to reduce the dimension while preserving the spatial characteristics, whereas the dense layers extract compressed non-linear features. After the first max-pooling layer we perform grouped convolutions – convolutions in parallel, wherein two identical CNNs (Conv-Net-1 and Conv-Net-2) are processed in parallel before their addition.

Then the decoder in the CNN-based AE, as shown in Fig. 9.4.1b, takes X as its input to reconstruct the 2D buildings $Y \in \mathbb{R}^{(500,40)}$. This is achieved by the dense layers that

decodes information from the compressed feature representation, and the convolutional and upsampling layers that reverts the building information to its original form.

We design a loss function $\mathcal{L}(\cdot)$ based on the Log-cosh loss instead of MSE to increase its robustness against outliers in the building data and also reduces the impact of appended zeros in the input \mathbf{I} . Note that the values in 2D matrix representation is standardized between 0 and 1 before feeding to the AE. For a grid (a, b) , we find the maximum and minimum values for that grid in all the training data and denote it by $\max(a, b)$ and $\min(a, b)$. Then we rescale all the values in the training and testing data for the $(a, b)^{\text{th}}$ grid as $(\text{value in } (a, b) - \min(a, b)) / (\max(a, b) - \min(a, b))$. The details of the designed loss function $\mathcal{L}(\cdot)$ and the CNN-based AE architecture can be found in Appendix 9.A.

We implement the AE in Keras [43] with TensorFlow as a backend. We keep the learning rate of 0.0012, batch size 16, and the total number of epochs 100. We train the network end-to-end using the Adam optimizer [42] over the time, to reconstruct the input 2D collapse of the building at the output of the decoder. Once the AE is converged³, we utilize the encoder to design compressed *Building* features $\mathbf{X} \in \mathbb{R}^{(12)}$ and then feed them to PL prediction.

9.5 ML-based Models and Training-Testing Methodology

We utilize the extracted *Clutter* and compressed *Building* features and compare the following regularized-linear and non-linear ML algorithms [41] for PL prediction using a street-by-street training and testing methodology to emphasize generalizability.

9.5.1 ML-based models for PL prediction

Let P denotes the true PL value, $\mathcal{F}_{(\cdot)}$ represents the input feature vector, and w indicates the designed weight matrix. The following ML algorithms are used for PL prediction.

1. *Lasso regression (Lasso)* optimizes the regression weights by minimizing the least-square error including a supplementary l_1 -norm penalty on the regression coefficients (weights),

$$\min_w \quad (1/2n_{\text{samples}}) \times \|\mathcal{F}_{(\cdot)}w - P\|_2^2 + \alpha^{l_1} \|w\|_1 \quad (9.2)$$

where n_{samples} is the sample size and $\alpha^{l_1} > 0$ imposes the l_1 penalty on the weights.

2. *Elastic-net regression (Elastic-net)* imposes both l_1 - and l_2 -norm penalties on the weights, where the convex combination of l_1, l_2 penalties is controlled by the pa-

³The designed AE with parallel Conv-nets (CNNs) converges after 50 epochs and has better reproducibility than using a single or two serially concatenated Conv-nets, shown in Appendix 9.B.

parameter $\alpha^{l_{12}}$,

$$\min_w \quad (1/2n_{\text{samples}}) \times \|\mathcal{F}_{(\cdot)}w - P\|_2^2 + \alpha^{l_1}\alpha^{l_{12}}\|w\|_1 + (\alpha^{l_1}(1 - \alpha^{l_{12}})/2) \times \|w\|_2^2 \quad (9.3)$$

3. *Random forest (RF)* is an ensemble learning method, where multiple decision trees' average is utilized to predict the PL. We consider 20 estimators with maximum tree depth of 25.
4. *Support vector regression (SVR)* solves the following primal problem

$$\begin{aligned} \min_{w,b,\zeta} \quad & (w^T w)/2 + C^{svr} \sum_{n=1}^T \zeta_n \\ \text{s.t.} \quad & P_n(w^T \phi(\mathcal{F}_{(\cdot)_n}) + b) \geq 1 - \zeta_n, \\ & \zeta_n \geq 0, \quad \forall n \in [1, T] \end{aligned} \quad (9.4)$$

where C^{svr} denotes the penalty term, ζ_n indicates the distance of n^{th} sample from the decision boundary, b represents the bias term and $\phi(\mathcal{F}_{(\cdot)_n})$ maps $\mathcal{F}_{(\cdot)_n}$ to a higher dimensional space. We train the SVR with an radial bias function (RBF) kernel, given by $\exp(-\gamma\|\mathcal{F}_{(\cdot)_i} - \mathcal{F}_{(\cdot)_j}\|^2)$ for any two samples i and j , and $\gamma > 0$.

All the PL prediction methods are implemented using scikit-learn [158]. We use grid search with 5-fold cross validation [41] over the training set to obtain the best parameters. In particular, the hyper-parameter α^{l_1} in Lasso and Elastic-net and C^{svr}, γ in SVR is grid-searched from $\{10^{-4}, 10^{-3}, \dots, 10^3, 10^4\}$ during the training and best fitted-parameter is used for testing.

9.5.2 Feature vectors

Point cloud-based expert street clutter features (Clutter) only

It consists of the seven expert features extracted from the street clutter information (in Sec. 9.3.2), represented as

$$\mathcal{F}_{\text{Clutter}} = \{\log\text{-}3D \text{ distance}, \log\text{-}1D \text{ distance}, 0D\text{-info}, \\ 3D\text{-info}, \text{street width}, \text{buildings on both sides}, Rx \text{ height}\} \quad (9.5)$$

Combination of Clutter and Building Features (Clutter + Building)

Herein we first use the encoder of the trained AE (proposed in Sec. 9.4) to obtain compressed features $\mathbf{X} \in \mathbb{R}^{(12)}$, and concatenate with the Clutter in (9.5), denoted by

Algorithm 1 Links-shuffle-split training and testing procedure

Require: $\mathcal{F}_{(\cdot)}$ and \mathbf{P} for 1028 PL measurements. \triangleright Collection of all the measurements from 13 streets.

Ensure: $\mathcal{D} = [\mathbf{x} := \mathcal{F}_{(\cdot)}, \mathbf{y} := \mathbf{P}]$. \triangleright This symbolises our total dataset.

- 1: **for** $i = 1$ **to** *iterations* **do** \triangleright Loop for varying shuffle and split of dataset.
 - 2: Randomly shuffle and split \mathcal{D} in 4 : 1 ratio to form $[\mathbf{x}_{\text{train}}^i, \mathbf{y}_{\text{train}}^i]$ and $[\mathbf{x}_{\text{test}}^i, \mathbf{y}_{\text{test}}^i]$.
 - 3: Train ML-based PL models (detailed in Section 9.5.1), using $[\mathbf{x}_{\text{train}}^i, \mathbf{y}_{\text{train}}^i]$ to obtain trained model \mathcal{M}^i .
 - 4: Test model \mathcal{M}^i using $\mathbf{x}_{\text{test}}^i$ to predict PL $\mathbf{y}_{\text{pred}}^i$.
 - 5: Calculate $RMSE^i$ in PL between ground-truth PL $\mathbf{y}_{\text{test}}^i$ and predicted PL $\mathbf{y}_{\text{pred}}^i$.
 - 6: **end for**
 - 7: Mean RMSE := $\mu(RMSE^{\{1,2,\dots,iterations\}})$ and std. deviation := $\sigma(RMSE^{\{1,2,\dots,iterations\}})$.
-

Algorithm 2 Street-by-street training and testing procedure

Require: $\mathcal{F}_{(\cdot)}^{\{1,\dots,13\}}$ and $\mathbf{P}^{\{1,\dots,13\}}$ for each of the 13 streets.

- 1: **for** $k = 1$ **to** 13 **do** \triangleright Loop for each of the 13 streets as testing.
 - 2: Testing Street := k^{th} street, Training streets := All the 13 streets **except** k^{th} street := $\{1, \dots, 13\} - \{k\}$.
 - 3: $\mathbf{x}_{\text{test}} := \mathcal{F}_{(\cdot)}^k$, $\mathbf{x}_{\text{train}} := \mathcal{F}_{(\cdot)}^{\{1,\dots,13\}-\{k\}}$, $\mathbf{y}_{\text{test}} := \mathbf{P}^k$ and $\mathbf{y}_{\text{train}} := \mathbf{P}^{\{1,\dots,13\}-\{k\}}$ \triangleright Measurements of k^{th} street becomes testing data, while all the other measurements form training data.
 - 4: Train ML-based PL models (detailed in Section 9.5.1), using $[\mathbf{x}_{\text{train}}, \mathbf{y}_{\text{train}}]$ to obtain trained model \mathcal{M}^k .
 - 5: Test model \mathcal{M}^k using \mathbf{x}_{test} to predict PL \mathbf{y}_{pred} .
 - 6: Calculate $RMSE^k$ in PL between ground-truth PL \mathbf{y}_{test} and predicted PL \mathbf{y}_{pred} for k^{th} testing street.
 - 7: **end for**
 - 8: Mean RMSE := $\mu(RMSE^{\{1,2,\dots,13\}})$ and std. deviation := $\sigma(RMSE^{\{1,2,\dots,13\}})$.
-

$$\mathcal{F}_{\text{Clutter.Buiding}} = [\mathcal{F}_{\text{Clutter}}, \mathbf{X}] \quad (9.6)$$

9.5.3 Training and Testing Methodology

In conventional ML-based training-testing, the 1028 PL measurements (collection of all the measurements from 13 streets) dataset is randomly shuffled and divided into 4 : 1 ratio for training and testing sets. We refer to it as *links-shuffle-split training and testing*, briefly described in Algorithm 1. To capture the impact of random shuffling and splitting, we perform the process multiple times and obtain mean RMSE and standard deviation in RMSE due to random shuffling and splitting of the dataset. Data in the testing set is statistically similar to those in the training set, and the focus of trained models is on interpolation. Since links that are close to each other have similar PL values, shuffling the data impacts negatively on the generalizability of the model given the limited amount of

PL measurements.

Motivated by the large street-by-street variation of measured PL observed from Manhattan measurements [128], we propose a new way of training-testing referred to as *street-by-street training and testing*, focusing on the extrapolation capabilities. We group the measurement links based on streets where they are collected and formulate 13 groups, one for each street. We then create 13 train-test combinations, wherein for each combination, one street is selected for testing and the rest for training. A model is trained and tested 13 times, using the 13 train-test combinations independently, producing 13 RMSE values. We summarize the procedure briefly in Algorithm 2.

Remark – In Algorithm 2 the street-by-street variation in prediction is quantified by the standard deviation in RMSE of the 13 tested streets. It is the metric chosen to measure the generalizability to unseen streets. Thus, the lower the standard deviation, the better is the generalizability of the proposed model to the unseen environment.

9.6 Performance Evaluation and Analysis

In this section, we evaluate the performance of the proposed PL prediction models using the *street-by-street training and testing*. The key performance metric is the mean and standard deviation of the 13 RMSE values obtained in street-by-street PL prediction. Our benchmarks are the 3GPP UMa LOS prediction model ($P = 28.0 + 22 \log_{10} d_{3D} + 20 \log_{10} f_c$), 3GPP UMi NLOS prediction model ($P = 22.4 + 35.3 \log_{10} d_{3D} + 21.3 \log_{10} f_c$), where $f_c=28$ denotes carrier frequency (in GHz), as well as the slope-intercept model described in (9.1) where the slope and intercept parameters are obtained using the same training data subsets as used by the ML-based methods. All of the three models only use the 3D Tx-Rx Euclidean distance d_{3D} as the input feature and their performance are evaluated using the same testing data subsets as used by ML-based methods.

9.6.1 PL Prediction Accuracy (RMSE)

We summarize in Table 9.5.1 the RMSE in PL prediction of linear and non-linear ML algorithms proposed in Sec. 9.5.1 with street-by-street training-testing methodology (shown in Algorithm 2), where the standard deviation of RMSE represents robustness against street-by-street variation over all 13 training-testing combinations.

The 3GPP UMi NLOS and 3GPP UMa LOS channel models have mean RMSE of 18.0 and 10.6 dB, respectively, not suitable for describing street canyon channels with clutters. The slope-intercept model produces a mean RMSE of 6.5 dB with a standard deviation of 2.0 dB across different testing streets. With the *Clutter* feature set ($\mathcal{F}_{\text{Clutter}}$), both regularized linear Elastic-net model and the non-linear SVR model simultaneously

Table 9.5.1: RMSE in PL prediction.

Prediction method	Street-by-street training-testing (mean \pm standard deviation) over 13 streets	Links-shuffle-split training-testing (mean \pm standard deviation) over 25 shuffles				
<i>3GPP UMi NLOS</i>	18.0 \pm 4.1 dB	18.1 \pm 0.4 dB				
<i>3GPP UMa LOS</i>	10.6 \pm 4.4 dB	11.3 \pm 0.4 dB				
<i>Slope-intercept</i>	6.5 \pm 2.0 dB	6.1 \pm 0.3 dB				
ML Algorithm	Clutter	Clutter + Building		Clutter	Clutter + Building	
		Average over 25 AE runs	Best out of 25 AE runs		Average over 25 shuffles	Best out of 25 AE runs
<i>RF</i>	6.6 \pm 1.8 dB	6.9 \pm 1.9 dB	5.8 \pm 1.5 dB	4.1 \pm 0.2 dB	4.3 \pm 0.2 dB	4.0 dB
<i>SVR</i>	5.4 \pm 1.4 dB	5.8 \pm 1.4 dB	4.8 \pm 1.1 dB	4.4 \pm 0.2 dB	4.4 \pm 0.2 dB	4.0 dB
<i>Lasso</i>	5.7 \pm 1.5 dB	5.7 \pm 1.4 dB	4.8 \pm 1.0 dB	5.0 \pm 0.2 dB	4.8 \pm 0.2 dB	4.4 dB
<i>Elastic-net</i>	5.4 \pm 1.3 dB	5.7 \pm 1.4 dB	4.8 \pm 1.1 dB	5.0 \pm 0.2 dB	4.8 \pm 0.2 dB	4.4 dB

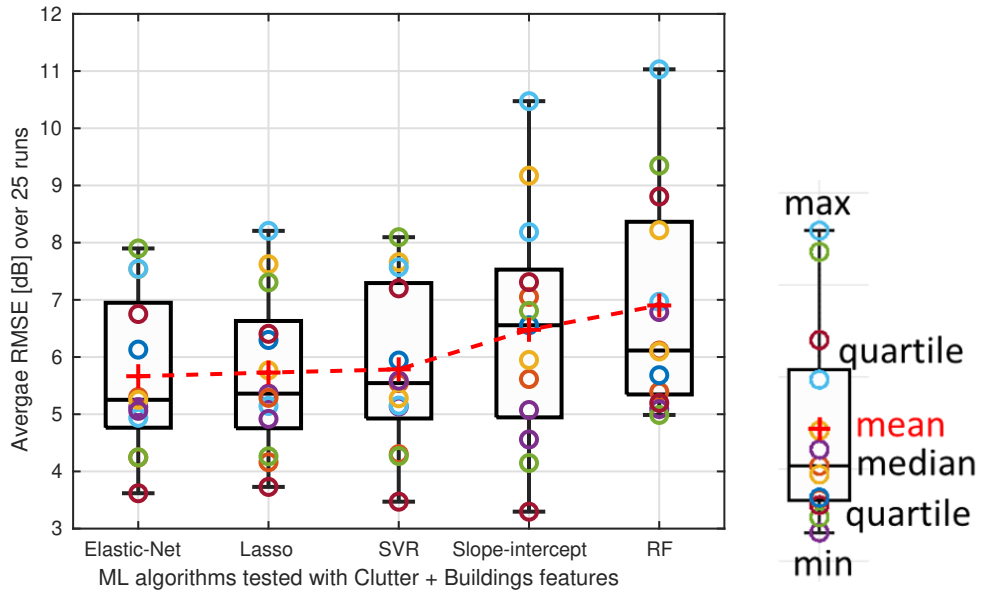


Figure 9.6.1: Street-by-street variation of the average RMSE over 25 runs using *Clutter + Building*.

reduce the mean RMSE by about 1.1 dB and street-by-street standard deviation by about 0.7 dB, creating a more generalizable model with better PL prediction accuracy.

The PL prediction performance can be further improved using the $\mathcal{F}_{\text{Clutter_Building}}$ feature set. Unlike the *Clutter* feature set which is deterministic, the $\mathcal{F}_{\text{Clutter_Building}}$ feature set contains compressed building features extracted from a CNN-based AE, which is inherently a random process and the resulting performance can change significantly [159]. We run the AE 25 times and generate 25 unique $\mathcal{F}_{\text{Clutter_Building}}$ feature sets. For each of the 25 feature sets, we test the ML algorithms using the street-by-street testing, reporting both the best⁴ and the average over all 25 runs. Street-by-street variation of the average RMSE over all 25 runs are reported in Fig. 9.6.1 using a box plot, where the average RMSE over 25 runs for each testing street is represented as a color-coded *o* symbol. The median and mean over 13 streets are given by line inside each box and the red + symbol, respectively, and the edges of the box mark the quartiles, with whiskers extending outside the box indicating the minimum and maximum over all 13 testing streets. By searching for a better AE out of 25 runs for each testing street, the mean RMSE can be further reduced by about 0.6 dB and street-by-street standard deviation by about 0.2 dB for both Elastic-net and SVR. The gain of Random Forest-based prediction over slope-intercept model is small, which is likely because it is not good at extrapolation when the statistics of the training and testing sets differ [160].

To verify and compare the capability of interpolation of various prediction models, we also run the classical links-shuffle-split training and testing approach detailed in Algo-

⁴Both the mean and standard deviation of the best RMSE converge within 25 runs as shown in Appendix 9.B.

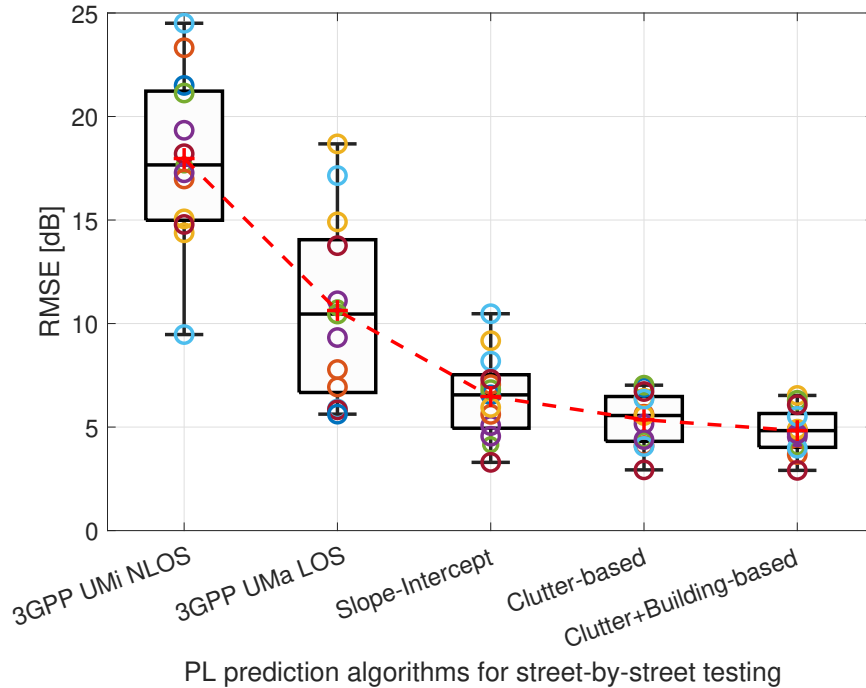


Figure 9.6.2: Street-by-street variation.

rithm 1 with $iterations=25$, i.e., randomly shuffle-and-split 25 times. The results are also shown in Table 9.5.1, where the 0.2 dB standard deviation in RMSE for *Clutter + Building* comes from both the inherent randomness of shuffling-splitting the dataset and AE compression out of 25 independent runs. The best result out of 25 runs for all the proposed ML-based PL prediction achieved over 1.7 dB reduction in mean RMSE as compared to the slope-intercept prediction, and over 0.4 dB reduction compared to the street-by-street testing. However, caution has to be taken to differentiate extrapolation and interpolation for site-specific PL prediction given limited measurement datasets.

9.6.2 Robustness Against Street-by-Street Variation and Distance

We evaluate the robustness of PL prediction against street-by-street variation in Fig. 9.6.2 using Elastic-net-based prediction for both the *Clutter* feature set and the best *Clutter + Building* ($\mathcal{F}_{Clutter_Building}$) feature set out of the 25 options. Compared to the two 3GPP models and the classical slope-intercept model, the two ML-based PL prediction models reduce both the mean RMSE as well as the street-by-street variability. This clearly demonstrates the importance of incorporating street-specific features such as street clutter and building into PL prediction models and the robustness of linear ML-based models in extrapolation to unseen streets.

The mean RMSE in PL prediction as a function of the 3D distance is shown in Fig. 9.6.3. All the links are grouped into 100 m intervals and within each interval we calculate the mean RMSE for street-by-street testing. Herein, X-axis denotes the ending

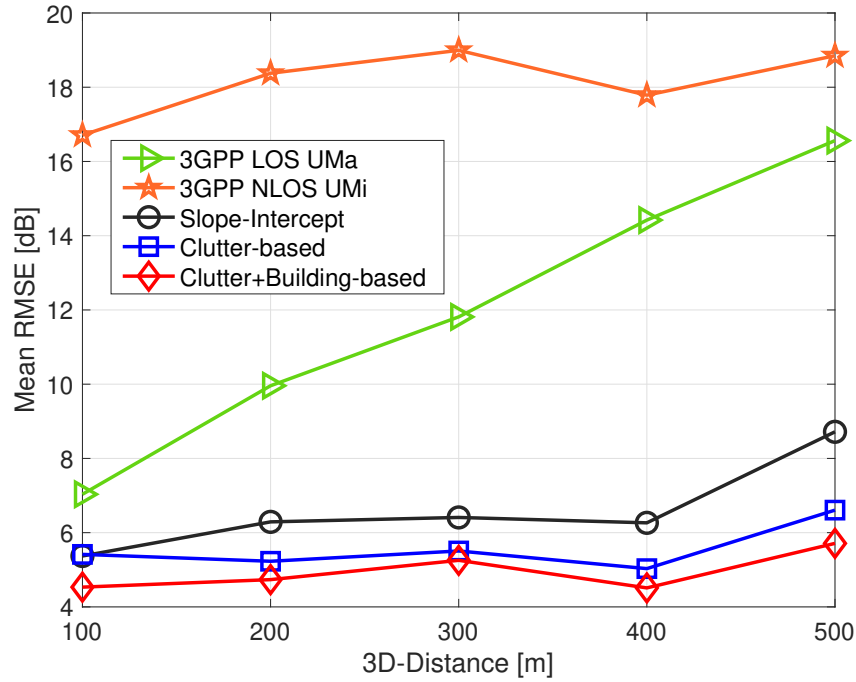


Figure 9.6.3: Mean RMSE versus distance.

position of an interval (e.g., 200 m denotes the interval spanning from 100 to 200 m). Our proposed models outperform the slope-intercept model for all the distance ranges. The improvement in mean RMSE is about 1.2 to 2.7 dB using *Clutter + Building* feature set compared to the classical slope-intercept model.

9.6.3 Feature Importance for the Point cloud-based Street Clutter Features

The Lasso regression imposes an l_1 -norm penalty that minimizes the weights of least relevant features to improve the accuracy, and therefore the relative magnitude of the resulting weights can be interpreted as the feature importance in PL prediction. We perform Lasso regression on the *Clutter* feature set with street-by-street training-testing methodology and present the obtained weights in Fig. 9.6.4, where each bar represents the mean value of the weights corresponding to each feature. The error bar indicates the minimum and maximum weight value obtained for that specific feature when tested separately for 13 streets. The amplitude of each weight indicates the importance of that feature, and the opposite signs (in amplitude) of the weights help balance the *Clutter* features in PL prediction. As expected, the *3D distance* has the highest weight, followed by the *0d-info*, which quantifies how cluttered each street is based on the normalized point cloud densities of the whole street. The '*Build. on both sides*' binary indicator and the *3d-info* which quantifies accumulated clutter density along the direct path also have notable im-

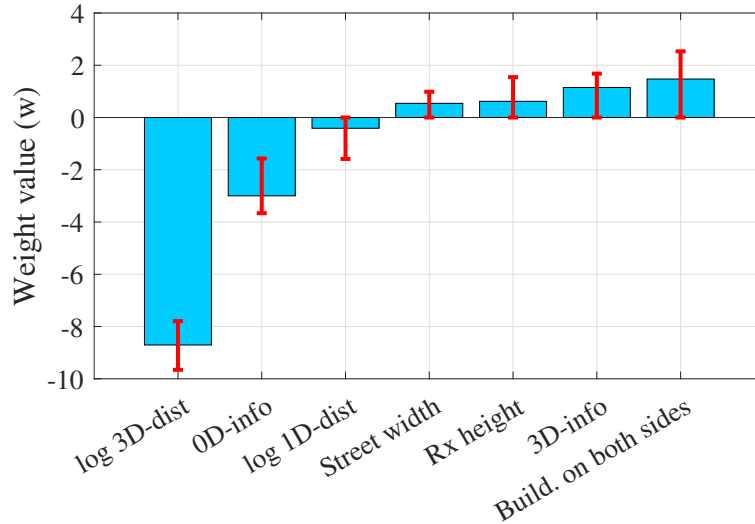


Figure 9.6.4: Average feature importance for *Clutter* feature set using Lasso regression in street-by-street testing, where the error bars indicate the minimum-maximum range over 13 streets.

Table 9.6.1: Training and testing complexity of PL prediction algorithms.

Algorithm	Training Complexity	Testing Complexity
<i>3GPP</i>	—	$\mathcal{O}(1)$
<i>Slope-intercept</i>	$\mathcal{O}(n)$	
<i>Lasso</i>	$\mathcal{O}(Tp^2n + p^3)$	$\mathcal{O}(p)$
<i>Elastic-Net</i>	$\mathcal{O}(Tp^2n + p^3)$	
<i>Random forest</i>	$\mathcal{O}(n \log(n)pn_t n_d)$	$\mathcal{O}(n_t n_d)$
<i>Support Vector</i>	$\mathcal{O}(n^2p + n^3)$	$\mathcal{O}(n_{svp})$

portance. This aligns with the understanding that reflection from buildings on both sides of the canyon increases signal strength, and that clutter intruded into the direct path has adversarial effect on propagation.

Feature importance can also be observed by comparing the change of prediction accuracy when each *Clutter* feature is excluded individually from prediction, as shown in Appendix 9.C using Elastic-net regression, where the same top four most important features are identified (in descending order): *0D-info*, *3D distance*, *3D-info*, and *building on both sides*.

By only using the top four most influential features, RMSE of 5.5 ± 1.1 dB can be achieved using Elastic-net regression. For almost all the ML-based prediction results using the reduced *Clutter* feature set, both the mean RMSE and standard deviations are within 0.2 dB from the results shown in Table 9.5.1 obtained using all the seven *Clutter* features. See Appendix 9.C for details.

9.6.4 Computational Complexity

Let us denote the number of training samples by n , number of features by p , number of trees by n_t and depth of the tree by n_d , and number of support vectors by n_{sv} , the training and testing complexity can be summarized as in Table IV where T is the number of outer iterations used in the coordinate descent solver. Given the low dimensionality of the features (7 in *Clutter* and 12 in *Building*) used in our PL prediction models, the PL prediction is very fast. The prediction time per link is less than $3 \mu s$ for Lasso and Elastic-net and less than $40 \mu s$ for RF and SVR when implemented using scikit-learn [158] running with Intel Core i7-6700 CPU with 64 GB RAM and Ubuntu 18.04.4 LTS OS.

9.7 Conclusions

We have proposed a ML-based PL prediction model for urban street canyon using the 28 GHz measurement data collected from Manhattan. The feature set contains street clutter obtained from LiDAR point cloud and buildings from 3D mesh-grid. The PL dataset has 1028 PL measurement links from 13 streets. Although the PL dataset is massive for the classical slope-intercept PL modeling, it is small for ML-based approaches when compared to massive point cloud and 3D building feature sets and parameters in AE and learning algorithms. To mitigate the risk of overfitting, we defined seven expert features with physical meaning from the point cloud. We also compressed 3D building features to a length-12 vector for each link using CNN-based AE. Instead of interpolation to nearby links, we focused on the extrapolation by introducing a street-by-street training and testing approach. Using linear ML algorithms for PL prediction, we achieved RMSE of 4.8 ± 1.1 dB compared to 10.6 ± 4.4 dB and 6.5 ± 2.0 dB for 3GPP LOS and slope-intercept prediction, respectively, which demonstrates the superior capability of our model in extrapolation.

Intuitive interpretation of feature importance was obtained using Lasso regression-based analysis and feature-exclusion analysis. By only using the top four most influential features, namely, distance, street clutter density (*0D-* and *3D-info*), and street canyon indication (*building on both sides*), RMSE of 5.5 ± 1.1 dB can be achieved using Elastic-net regression.

Among the four learning algorithms used in this paper, the non-linear Random Forest regression has achieved the worst prediction performance under street-by-street testing but is the best under links-shuffle-split testing. This may be attributed to the reduced similarity in statistics between training and testing sets in street-by-street testing compared to the links-shuffle-split testing, and the lack of extrapolation capabilities of Random Forest regressions. Regularized linear Elastic-net regression has the best performance, which is in line with the intuition that regularization on linear algorithms is more robust against

overfitting with limited training dataset. The non-linear Support Vector Regression with radial bias function kernel performs well (second best) for both links-shuffle-split testing and street-by-street testing, at the cost of higher computational complexity.

Appendix

9.A CNN-based Autoencoder design to Compress Building Features

We design CNN-based AE for feature extraction from the building dataset, as described in Fig. 9.3.2, where an encoder $e(\cdot)$ compresses the input 2D building collapse $\mathbf{I} \in \mathbb{R}^{(500,40)}$ to an representation $\mathbf{X} \in \mathbb{R}^{(12)}$, which is then fed to the decoder $d(\cdot)$ to reconstruct original input 2D building collapse \mathbf{I} . We briefly describe below the network layers shown in Fig. 9.4.1.

1. *1D-convolutional layer (Conv 1D)* - It employs various kernels to convolve the 2D-image, preserving the spatial characteristics of the input image while extracting relevant features.
2. *1D-max-pooling layer (Max-Pooling 1D)* - Pooling is a sample-based discretization process utilized to downsample the input image by making assumptions in the binned sub-region. In max-pooling, we take the maximum value in the binned sub-region.
3. *1D-upsampling layer (Up-Sampling 1D)* - Upsampling layer has no weights, which helps increase the input dimensions when followed by a convolutional layer.
4. *Fully connected layer (Dense)* - The non-linear processing is performed via dense layers wherein each neuron is fully connected to all the neurons in the previous layer.

Please note that 1D in the above layers means we have kernels (in Conv layer) and factor (for upsampling and downsampling) [41] in only one-dimension.

9.A.1 Designing the Encoder

The input to our network is 2D collapse of buildings given by $\mathbf{I} \in \mathbb{R}^{(500,40)}$, which is normalized between 0 and 1. Let $e(\mathbf{I}|\theta_e)$ be the mapping from the input buildings to

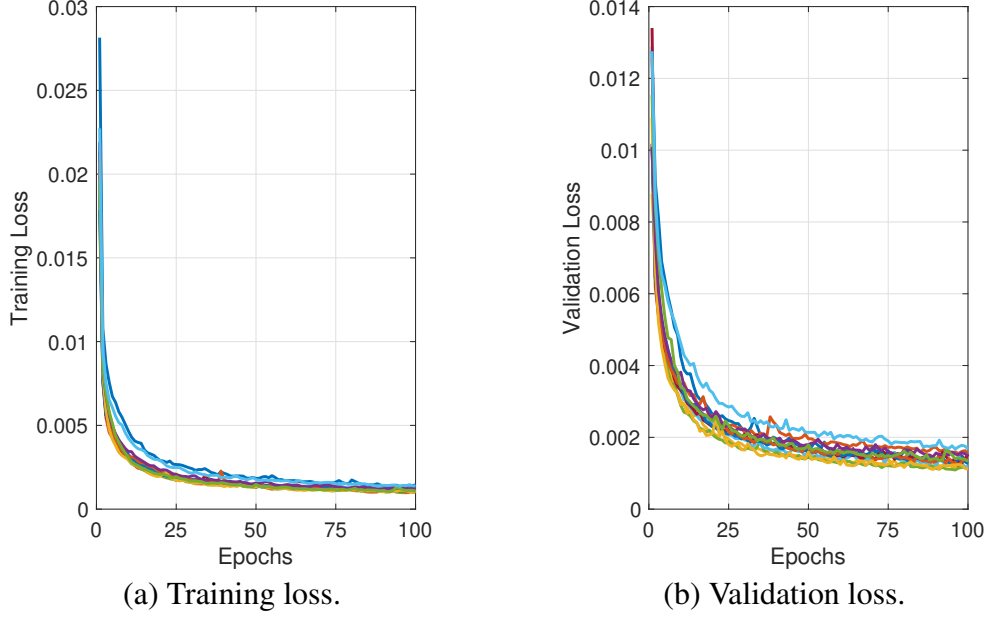


Figure 9.A.1: Convergence of CNN-based AE for 13 testing streets.

compressed representation when the parametric transformation of encoder is given by θ_e , which denotes the weight, filters, and bias terms. Thus the encoder can be denoted as

$$\mathbf{X} = e(\mathbf{I}|\theta_e) = e_{L_e}(\dots(e_2(e_1(\mathbf{I}|\theta_{e_1})|\theta_{e_2})\dots|\theta_{e_{L_e}})) \quad (9.7)$$

where L_e denotes the number of layers in the encoder. For the convolutional layers, the operation of the l^{th} layer can be represented as follows:

$$\mathbf{X}_l = e_l(\mathbf{I}_l|\theta_{e_l}) = h(\mathbf{W}_l \otimes \mathbf{I}_l + b_l), \quad \theta_{e_l} = [\mathbf{W}_l, b_l] \quad (9.8)$$

where \otimes indicates the convolutional process, \mathbf{W}_l represents the 1D kernels used for feature extraction, b_l denotes the bias vector, $h(\cdot)$ is the activation function, $\mathbf{I}_l = \mathbf{X}_{l-1}$ comes from layer concatenation and \mathbf{I}_1 equals the 2D building matrix \mathbf{I} .

We apply several *Max-Pooling 1D* layers in between for improving the region covered by the following receptive fields. Moreover, as shown in Fig. 9.4.1a, we introduce grouped convolutions in the encoder, wherein we take the output of the first *Max-Pooling 1D* and make two branches of it, with separate *Conv 1D* and *Max-Pooling 1D* layers, and then add the output of both branches (inspired by AlexNet [41]). The convolutional layer's output is then flattened to \mathbf{K} and used as input of several stacked dense layers, where the first dense layer in the encoder can be given by

$$\mathbf{X}_l = e_l(\mathbf{I}_l|\theta_{e_l}) = h(\mathbf{W}_l \mathbf{K} + b_l), \quad \theta_{e_l} = [\mathbf{W}_l, b_l] \quad (9.9)$$

9.A.2 Designing the Decoder

The input to our decoder is the output of the encoder given by $\mathbf{X} \in \mathbb{R}^{(12)}$. Let $d(\mathbf{X}|\theta_d)$ be the mapping from the compressed representation to input buildings when the parametric transformation of decoder is given by θ_d , which denotes its weight, filters, and bias terms. The decoder can be represented as follows:

$$\mathbf{Y} = d(\mathbf{X}|\theta_d) = d_{L_d} \left(\dots (d_2 (d_1 (\mathbf{X}|\theta_{d_1}) |\theta_{d_2}) |\theta_{d_{L_d}}) \right) \quad (9.10)$$

where L_d denotes the number of layers in the decoder. It performs a reverse operation of encoder here to generate the output $\mathbf{Y} \in \mathbb{R}^{(500,40)}$ of the same size as the input \mathbf{I} .

As shown in Fig. 9.4.1, we use the *Tanh* activation function $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ for all the layers because *Tanh* activation function performed the best compared to the other non-linear activation functions, except for the last layer, where we used the *ReLU* activation function $\text{ReLU}(x) = \max(0, x)$ to ensure a positive real-value output. Moreover, $a \times b$ on each *Conv 1D* layer indicates the filters and kernel size. The value on each *Max-Pooling 1D* and *Up-sampling 1D* denotes the factor by which downsampling and upsampling are performed on the first dimension. Also, value on each *Dense* layer indicates the number of neurons considered in that layer. The symbol $+$ indicates the addition of the outputs of two previous layers.

9.A.3 Designing the Loss Function

We use log-cosh loss, which is the logarithm of the prediction error's hyperbolic cosine. Also, we have \mathbf{I} as the input to the encoder in the AE as well as the ground truth to be predicted from the decoder and \mathbf{Y} as the predicted output of the AE. Thus the difference between the input and output of the AE can be given by $\varphi_{i,j} = \mathbf{Y}_{i,j} - \mathbf{I}_{i,j}, \forall (i, j)$, where $i = \{1, \dots, 500\}$ and $j = \{1, \dots, 40\}$ denotes the length and width of the streets (with building facades). We choose log-cosh loss to help stabilize the training performance with fewer epochs (iterations) because the outliers minimally impact the log-cosh loss compared to the MSE loss [41]. Also, since we have appended zeros in the input \mathbf{I} , there have many zeros appended for shorter distances, which makes it difficult for the AE network to learn non-zeros values in closer distance ranges. Thus, we introduce a matrix $\hat{\mathbf{Y}} \in \mathbb{R}^{(500,40)}$, where for the n^{th} training sample, given by

$$\hat{\mathbf{Y}}_{i,j}^n = \begin{cases} 0, & \text{if } \mathbf{I}_{i,j}^n = 0 \\ \mathbf{Y}_{i,j}^n, & \text{otherwise} \end{cases}, \quad \forall (i, j) \quad (9.11)$$

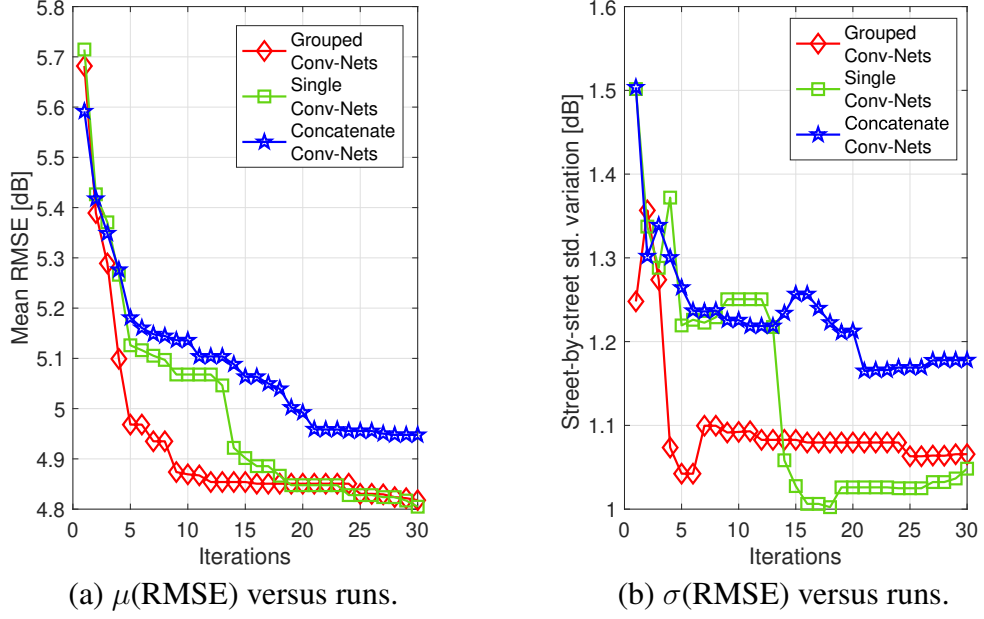


Figure 9.B.1: Evaluation of grouped CNNs and iterations needed for reproducibility.

Then the combined loss function for the n^{th} training sample can be given by

$$\mathcal{L} = \mu \left(\log \left(\cosh \left(\hat{\mathbf{Y}}^n - \mathbf{I}^n \right) \right) \right) + 0.1 \times \mu \left(\log \left(\cosh \left(\mathbf{Y}^n - \mathbf{I}^n \right) \right) \right) \quad (9.12)$$

where $\mu(\cdot)$ is the mean. The loss function has two parts, the first part focuses on the reconstruction error of the non-zero values. The second part focuses on the reconstruction error of all the values, whereas the weight 0.1 helps us in reducing the impact of the appended zeros.

9.B Loss Convergence and Reproducibility of the Proposed CNN-based AE

We utilize the designed AE to extract the compressed feature vector $\mathbf{X} \in \mathbb{R}^{(12)}$ from the 2d collapse $\mathbf{I} \in \mathbb{R}^{(500,40)}$ of the 3D building dataset. Then we train the AE in an end-to-end manner by minimizing the designed loss in (9.12). The convergence of the training and validation losses for 13 models created for street-by-street training and testing are shown in Fig. 9.A.1, wherein the loss converges within 50 epochs (iterations utilized by Adam optimizer to converge).

In Fig. 9.B.1, we evaluate the compressed features (\mathbf{X}) by making three types of AEs, wherein *only* the grouped CNNs in the encoder as proposed in Fig. 9.4.1a is replaced by: (1) Grouped Conv-Nets – as proposed, (2) Single Conv-Net – remove *Conv-Net-2* from

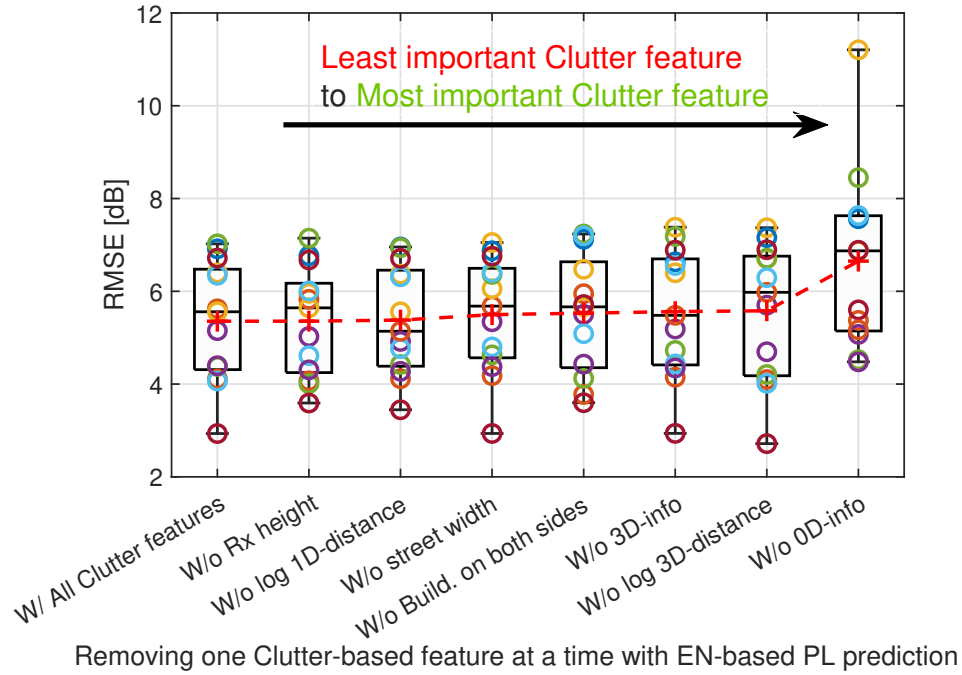


Figure 9.C.1: Importance of each feature in Elastic-net-based PL prediction model.

the encoder, and (3) Concatenate Conv-Nets – concatenate *Conv-Net-1* and *Conv-Net-2* serially, to obtain their respective feature set $\mathcal{F}_{\text{Clutter_Building}}$. Further, we show the best RMSE performance achieved by the Elastic-net regression to predict the PL, with a varying number of iterations. Grouped CNNs perform the best with smoother convergence compared to others. Furthermore, Fig. 9.B.1 also shows that with 25 iterations we can achieve the *reproducibility* for the best PL prediction RMSE performance.

9.C Analyzing the importance of designed Clutter in PL Predictions

In Fig. 9.6.4, we utilized l_1 -norm based Lasso regression to determine the importance of the individual *Clutter* feature if all the *Clutter* features (1 to 7) are provided for the PL prediction. The importance of features, as quantified by their Lasso weights, in the descending order is given as: *log 3D-distance*, *0D-info*, *building on both sides*, *3D-info*, *Rx height*, *street width*, and *log 1D-distance*.

Feature importance can also be observed by comparing the change of prediction accuracy when each *Clutter* feature is excluded individually from prediction. This approach works for all ML-based prediction algorithms and thus provides an alternative way of assessing feature importance.

In Fig. 9.C.1, we remove one feature at a time from the 7 *Clutter* features and determine the RMSE in Elastic-net based PL prediction using street-by-street testing in

Algorithm 2. Removing the *OD-info* feature has the strongest consequence, degrading the mean RMSE by as much as 1.3 dB. Therefore, based on the degradation of mean RMSE, the importance of the features⁵ in the descending order can be given as: *OD-info*, *log 3D-distance*, *3D-info*, *building on both sides*, *street width*, *log 1D-distance*, and *Rx height*.

Based on observations from Fig. 9.6.4 and Fig. 9.C.1, we conclude that the most important features are the *log-3D-distance*, street-clutter information given by *OD-info* and *3D-info*, and canyon status (*buildings on both sides*), referred as *Clutter₄*. We analyze the RMSE in PL prediction using *Clutter₄* for both street-by-street and shuffle-split testing, shown in Table 9.A.1. By only using the top four most influential features, RMSE of 5.5 ± 1.1 dB can be achieved. When using *Clutter₄*, both the mean RMSE and standard deviations of almost all the ML-based predictions are within 0.2 dB from the results where all the 7 clutter features are used.

⁵The potential correlation among multiple features may underestimate the importance of a feature if it has high correlation with others.

Table 9.A.1: RMSE in PL prediction using the 4 most important Clutter features ($Clutter_4$).

ML Algorithm	Street-by-street training-testing		Links-shuffle-split training-testing	
	$Clutter_4$	$Clutter_4$ + Building Average over 25 AE runs	$Clutter_4$	$Clutter_4$ + Building Average over 25 AE runs
<i>RF</i>	6.2 ± 1.4 dB	6.9 ± 1.9 dB 5.8 ± 1.5 dB	4.1 ± 0.2 dB	4.3 ± 0.2 dB 3.9 dB
<i>SVR</i>	5.6 ± 1.2 dB	5.7 ± 1.3 dB 4.8 ± 0.9 dB	4.3 ± 0.2 dB	4.3 ± 0.2 dB 4.0 dB
<i>Lasso</i>	5.7 ± 1.5 dB	5.7 ± 1.3 dB 4.8 ± 1.0 dB	5.1 ± 0.2 dB	4.8 ± 0.2 dB 4.5 dB
<i>Elastic-net</i>	5.5 ± 1.1 dB	5.5 ± 1.2 dB 4.7 ± 1.0 dB	5.1 ± 0.2 dB	4.8 ± 0.2 dB 4.5 dB

Chapter 10

Conclusions and Future Directions

10.1 Summary

This work focused on the learning-based communication system design by employing autoencoder (AE) frameworks.

10.1.1 Summary of Various Proposed AE frameworks

We summarize various kinds of AE frameworks proposed in this work in Fig. 10.1.1. In particular, we studied the Channel AE framework (that adds redundancy) for high-dimensional block coded modulation (BCM) design with the channel state information (CSI) knowledge and differential BCM (d-BCM) design without the CSI knowledge in Chapters 2–8, for high transmission rates $R = k/n$ [bits/channel-reuse] and short block lengths (n), and the Vision AE framework (that removes redundancy) for high-dimensional feature compression for machine learning (ML)-based mm-wave path-loss (PL) predictions in Chapter 9.

This work began by studying the AE-based BCM and d-BCM design in point-to-point (P2P) communication networks in Chapter 2. We classify the AE frameworks as Fixed AE and Variable AE, depending on if the number of neurons in the dense layers remains independent and dependent on the input-output bits (k), respectively. Based on the number of optimized parameters and bit-error-rate (BER) performance, we showed that the Variable AE and Fixed AE frameworks are suitable for low rate $R \leq 8/7$ and high rate $R > 8/7$, respectively. Moreover, the Fixed AE facilitates designing a single neural network (NN) architecture for varying rates R . Thus, the Fixed AE framework is employed in this thesis.

Then, we studied the AE-based BCM and d-BCM designs for the amplify-and-forward (AF) relaying networks in Chapter 3–6. In Chapter 3, depending on the input-output of the AE and minimized cross-entropy (CE) loss, we classified the AE frameworks as the

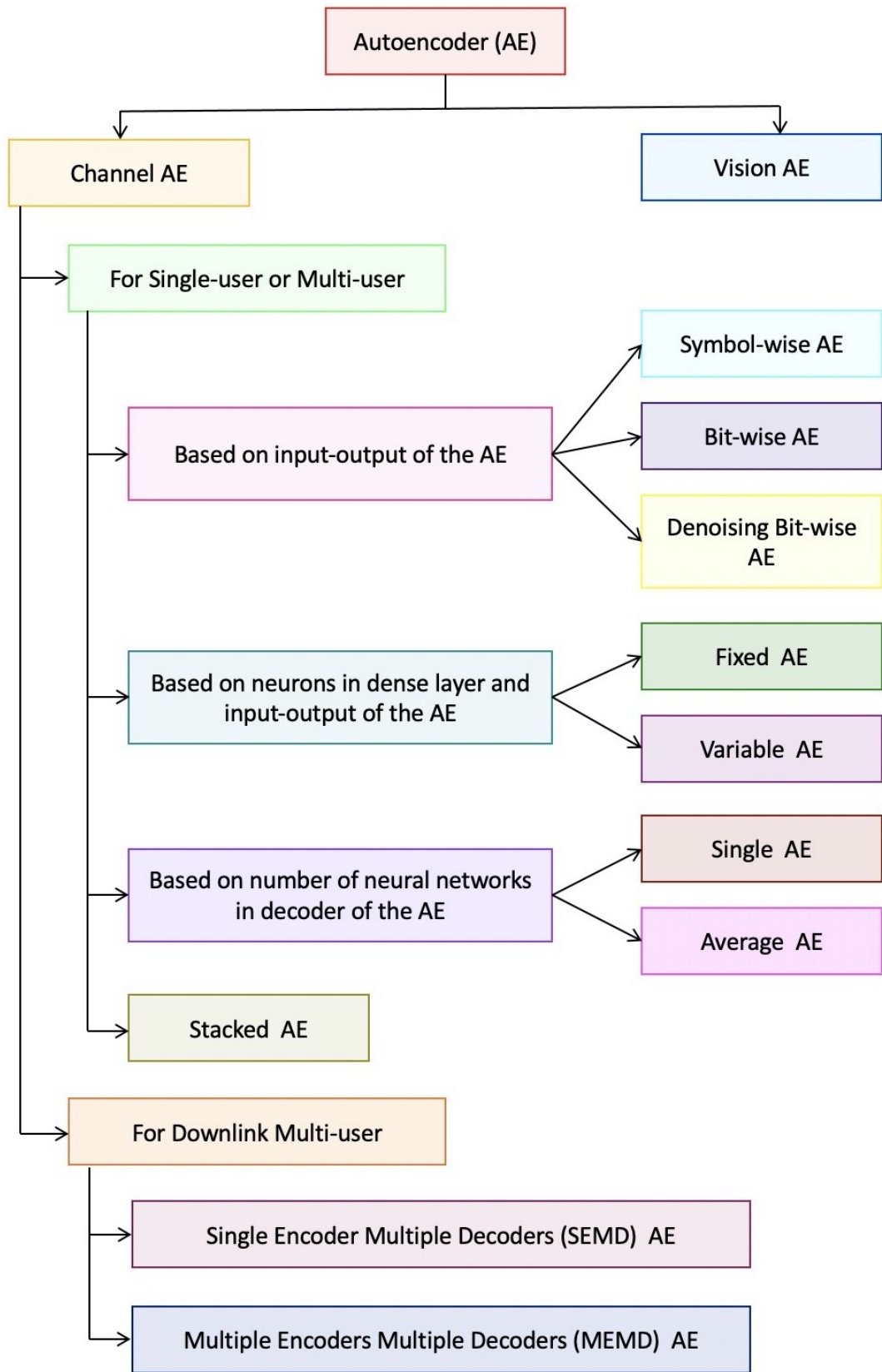


Figure 10.1.1: Summary of various kinds of AE frameworks proposed in this work.

bit-wise AE and symbol-wise AE frameworks. We show that although the bit-wise and symbol-wise AEs almost forms similar (differential) block-coded modulation designs. However, the bit-wise AE showed three distinct advantages over the symbol-wise AE – (1) automatic bit-labeling, possibly in a Gray-coded format, (2) faster training convergence at low signal-to-noise-ratio (SNR) regimes, and (3) reduced training–testing time and implementation complexity. Thus, we proposed to employ a bit-wise AE framework in this thesis. Furthermore, in Chapter 7, we propose a novel bit-wise denoising AE for the BCM and d-BCM design in a decode-and-forward (DF) relaying network. In particular, the bit-wise denoising AE takes soft-probabilistic values as input at the encoder side (instead of bits as in the bit-wise AE framework) and aims to denoise these soft probabilities to intended bits at the decoder side, by utilizing a novel proposed training policy.

In Chapter 4, based on the number of neural networks (NNs) in the decoder of the AE framework, we propose a novel Average AE framework with three small NN decoders instead of a Single AE, as proposed in the rest of the thesis that employs a single large NN decoder. The key benefits of the proposed Average AE over Single AE are two folds – (1) significantly reduced optimization parameters in the NN decoder, and (2) the averaging from multiple soft outputs helps improve the bit-error-rate (BER) performance.

We studied AE-based BCM and d-BCM designs in a DF relaying network in Chapter 7. We proposed a novel stacked AE framework with new training policies. In particular, we propose a stacked AE-based BCM and d-BCM designs for the one-way DF (OW-DF) and two-way DF (TW-DF) relay networks. For the AE-based OW-DF relay network, we propose to employ a bit-wise AE in the first transmission phase and stack a bit-wise denoising AE for the second phase transmission. For the AE-based TW-DF relay networks, we propose to employ a bit-wise AE framework trained for the P2P networks, in a stacked form to make a TW-DF relay network. We show that the stacked AE framework provides a method to train the AE framework for DF relaying protocol and also reduces the NN optimization parameters drastically while improving the BER performance.

In Chapter 8, we studied AE-based BCM and d-BCM designs for a downlink multi-user network, where a base station (BS) intends to communicate with multiple users by multiplexing them together on the same resource block in the power domain. Each user has a separate NN-based decoder, while based on the number of NN encoders at the BS we classify the AE frameworks as single encoder multiple decoders (SEMD) and multiple encoders multiple decoders (MEMD) frameworks. We showed that the SEMD AE outperforms the MEMD AE framework even with reduced NN optimization parameters, because of the optimal superposition of signals of multiple users.

10.1.2 Special Notes for Channel AE frameworks

Loss Optimization

In this thesis, we perform end-to-end training while minimizing the total binary CE loss for all the nodes ‘together’ in an AE framework by solving a multi-label binary classification problem. For example, in Chapter 5, we minimize the total binary CE losses of both the terminal nodes together, which enabled us to tackle the interference (at the relay node) and remove the deteriorating impacts of hardware impairments. Moreover, in Chapter 8, we minimize the total binary CE losses of all the multiple downlink users together, which enables us to tackle the interference from multiple users.

Furthermore, in Chapter 8, we also propose a novel methodology to impose the quality-of-service (QoS) constraints in the AE frameworks. Specifically, we propose to employ a scaling factor in the loss minimization that enables to impose of the QoS constraints and propose a scaling factor updating policy. This is especially advantageous for internet-of-things (IoT) networks where we need to impose strict QoS constraints.

We show that the optimization problems of the proposed AE framework and the traditional communication networks obtain the same encoder-decoder and bit-labeling when the AE is trained with bits as input-output, Sigmoid activation at the last layer of the decoder, and binary CE loss is optimized.

Also, we show that by minimizing the binary CE loss we are inherently maximizing the bit-wise mutual information (MI) for all the nodes together and the generalized MI of the network.

Training Convergence

We focus on the training convergence of the proposed AE frameworks from an information-theoretic perspective, as follows:

- We analyze the convergence of the proposed AE frameworks by estimating the mutual information (MI) between the input-output of the AE framework, specifically
 - We analyze the convergence for varying SNR levels and/or RSI levels (in a full-duplex network). We show that, for a sufficiently large block length, the training of the AE framework converges to a global minima above a minimum required SNR levels and/or below a maximum RSI level.
 - We compare the convergence of the AE-based BCM and d-BCM designs. We show that the CSI knowledge is helpful to reach the global minima faster.
 - We compare the convergence of various AE frameworks, such as the Fixed and Variable AEs, or the symbol-wise and bit-wise AE frameworks.

- For any given hyper-parameter settings and rate $R = k/n$, with sufficiently large block length (n), we provided the two necessary conditions for the training convergence of the AE frameworks performing BCM or d-BCM designs, detailed as follows:
 - The validation CE loss of the AE framework has converged.
 - The NN encoder of the AE designs 2^k codewords.
- We show that the BER performance of the AE framework performing BCM design converges with at least 2^{k+1} training samples, while that for d-BCM design converges with at least 2^{k+5} training samples. Also, we show that the full-duplex transmission mode requires double training samples for the convergence of the BER performance of the AE framework as compared to the half-duplex transmission mode because of the RSI.

NN Architecture Designs

In this thesis, we always aim to create a generalizable NN architecture that can perform either BCM or d-BCM designs with minimal changes in the NN architecture. Thus, we employ the CSI knowledge in the Lambda layer L_L of the NN decoders to differentiate in the NN architectures of the AE frameworks for performing BCM or d-BCM designs. Hence, throughout this thesis, only the Lambda layer L_L is modified in the NN architecture for performing BCM and d-BCM design in the AE framework.

We design and develop the NN-based radio transformer network (RTN) in the Lambda layer L_L of the NN-based decoder. Moreover, we note that RTN can have either 2 outputs representing the equivalent channel or $2n$ outputs representing the equalized output by the NN. In Chapter 3, we showed that including an RTN in the NN-based decoder doesn't impact the BER performance but helps in the faster training convergence of the AE frameworks. Furthermore, we showed that the RTN helps in AE-based d-BCM design for half-duplex networks¹ and AE-based BCM design in the full-duplex networks².

In Chapter 5, we showed that until complex-valued NNs are utilized, it is (always) best to perform complex domain operations in Lambda layers and is also (sometimes) suitable to concatenate the complex-values information as real-valued together. Thus in this thesis, for the AE-based BCM designs, we always perform channel equalization and zero-forcing (ZF)-based in-phase and quadrature-phase imbalance (IQI) compensation in the Lambda layers.

In the NN architecture designs of the AE frameworks, we note that the dense layers are beneficial than the CNN layers, the *Tanh* activation function is better in the dense

¹Chapters 2, 3, 4, 5, 7, and 8.

²Chapter 6.

layers, and the use of step-decay learning rate scheduler and early stopping improves the AE's performance and reduces the overfitting.

Generalizability

In this thesis, we create a single AE framework performing BCM or d-BCM design that can be deployed for any testing SNR. For this, we proposed to create a training dataset of a large number of block samples for each of the SNR values. We show that for additive white Gaussian noise (AWGN) channels, training only one low SNR point brings the best AE performance, and for the Rayleigh block fading (RBF) channels, we need multiple SNR points for training the AE framework to obtain its best performance, ranging from low SNR to high SNR points. Furthermore, we note that for AE-based modulation design, training on a single low SNR point also brings the best AE performance in RBF channels.

In Chapters 4, 5, 7, we proposed AE frameworks with IQI at various nodes. In addition to generalizability towards testing SNR, we note that the generalizable AE frameworks can be trained for removing the deteriorating impacts of various amplitude offsets or phase offsets in Chapters 4, 5. Furthermore, in Chapter 7, we show that we can train a single AE framework that can be generalizable to all the varying testing SNR and phase offsets and amplitude offsets together. For this, we create a training dataset of a large number of block samples for multiple SNR values (as detailed above). Further, for each of the SNR values, we also include training samples from high phase and amplitude offsets to low phase and amplitude offsets. This helps the training dataset to encompass all the high and low IQI for each SNR in the training of the AE, making a single trained AE framework robust to varying SNR and IQI.

Similarly, in Chapter 6, we design a single AE framework that can generalize well on varying testing SNR or residual self-interference (RSI) levels in the full-duplex amplify-and-forward (FD-AF) relay networks.

The aforementioned method of creating a training dataset leading to this generalizability is not limited to a single user scenario but is also applicable to multiple user scenarios as detailed in Chapter 8.

Reproducibility

We analyzed the reproducibility of the AE framework by varying training-validation data and weight initialization for training each of the AE performing BCM and d-BCM designs 25 times and reporting the standard deviation and mean of BER in testing data. We note that the proposed AE frameworks are highly reproducible because their standard deviation of 25 BER obtained from 25 different runs lies in the range $10^{-2} - 10^{-4}$, depending on the distortion levels in the communication networks.

Interpretability

In this thesis, we focus on the interpretability of the AE-based BCM and d-BCM designs. In Chapter 2, we propose the distinct properties of the AE-based BCM and d-BCM designs for (n, k) block or rate (R) , detailed as follows:

1. Fixed AE framework designs 2^k codewords in $2n$ -dimensional space.
2. The minimum Euclidean distance between any of the possible codewords increases as the rate decreases.
3. The minimum Euclidean distance between any of the possible codewords is greater for BCM design with the CSI knowledge compared to the d-BCM design without the CSI knowledge.
4. For unit block length, the scenario of only AE-based modulation design, the minimum Euclidean distance between any of the possible codewords reduces to zero for the d-BCM designs without the CSI knowledge.
5. The AE-based designed BCM and d-BCM reaches its maximum potential of decoding (in terms of BER performance) when $n = \lceil k/2 \rceil$ and $n = \lceil (k + 1)/2 \rceil$, respectively.
6. The minimum Euclidean distance between any of the two closest codewords varies largely for the d-BCM design than for the BCM design.
7. When the number of codewords becomes extremely large, the minimum Euclidean distance between any of the two closest codewords follows a Gaussian distribution for sufficiently large block length.
8. As the block length increases, the Euclidean distance between any of the two closest codewords concentrates to the average Euclidean distance.
9. The packing density improves as the rate decreases.
10. The packing density of the BCM design with the CSI knowledge is better than the d-BCM design without the CSI knowledge.
11. For sufficiently large block length, the packing density of the BCM and d-BCM designs converge.
12. The codewords designed by the AE framework are spherical codes.

We also evince the applicability of the aforementioned properties of the AE-based framework in a much complicated full-duplex relaying network in Chapter 6.

In Chapter 8, we analyze the multi-user codeword mapping. Specifically, we utilize the t-stochastic neighbor embedding (t-SNE) algorithm to visualize this $2n$ -dimensional data in 2-dimensional space, and then we utilize the k -medoids to cluster the t-SNE results into k subsets and finally map the bits transmitted with each cluster to obtain the codeword mapping.

Throughout the thesis, we propose to utilize metrics such as minimum Euclidean distance, normalized second-order and fourth-order moments, and constellation figures of merit to analyze the BER performance gains achieved by the proposed AE-based framework over the traditional methods.

AE's Performance

In this thesis, we performed AE-based BCM and d-BCM design for short block length ($n = 7$) and high transmission rates R . For a conventional network, we consider conventional (differential) modulation techniques with maximum likelihood decoding (MLD), and also $(7, 4)$ Hamming code as baseline error-correction codes. We show that the proposed AE-based BCM and d-BCM designs outperform the conventional networks and note the following two key observations:

- The BER performance gains of the AE frameworks increase with the rate R .
- The BER performance gains of the AE-based d-BCM designs (without the CSI knowledge) is greater than AE-based BCM designs (with the CSI knowledge), compared to the conventional networks.

Interestingly, we note that as the complexity of the communication network increases, the BER performance gains of the proposed AE-based BCM and d-BCM designs increases. This is because the traditional modulation and error correction techniques are not designed optimally and specifically for the complex communication networks, further, in scenarios such as with hardware impairments, channel estimation errors, and residual interference, because of the lack of mathematical models in traditional methods. This can be seen in multiple chapters, detailed as follows:

- In Chapters 4, 5, 7, we proposed AE frameworks with hardware impairments (IQI) at various nodes. Within the proposed AE frameworks, we propose a BCM design with ZF-based IQI compensation that considers the availability of the CSI and IQI knowledge. However, the IQI and CSI need to be estimated separately, increasing the feedback overhead. Thus, we also propose a BCM design with no IQI compensation that requires only the CSI knowledge. Finally, we propose a d-BCM design that removes the necessity of both the CSI and IQI knowledge. Further, we analyze the performance under low signal-to-interference-and-noise-ratio regimes and

show that proposed AE frameworks can almost completely remove the IQI, thereby completely removing the error floor and outperforming the MLD of the traditional networks with no hardware impairments (IQI). Thus, the proposed AE framework reduces the feedback overhead in IQI/CSI estimation and improves the BER performance gains extraordinarily, especially for differential scenarios.

- In Chapter 6, we proposed AE-based BCM and d-BCM design for the full-duplex AF (FD-AF) relay networks in the presence of the residual self-interference (RSI). We show that as the modulation order increases the proposed AE can even outperform the conventional half-duplex AF relay networks even for higher RSI. Furthermore, although the traditional FD-AF relay networks can not decode the signals in the presence of high RSI, the proposed AE frameworks can decode the signals for any RSI, showing remarkable BER performance gains.
- We analyze the AE's performance in the presence of channel estimation error. We note that the proposed AE framework with almost entirely erroneous channel estimation still outperforms the conventional networks with perfect channel estimation. This is because we train the single AE framework on various channel estimation errors, such that it can generalize well on varying channel estimation qualities.

Thus, as the imperfections in the communication networks increase the BER performance gains of the proposed AE frameworks increases. This shows the capabilities of the AE framework in designing future networks and tackling the non-linear distortions and imperfections in IoT devices.

As the number of users increases in the network, traditionally it becomes increasingly difficult to perform the BCM design and unsolvable to perform d-BCM design. In Chapter 8, we design AE-based BCM and d-BCM design for downlink multi-user networks, considering the conventional non-orthogonal multiple access (NOMA) as a benchmark. In contrast to the NOMA, we remove the successive interference cancellation (SIC) at the strong users and the requirement of the power allocation factor (PAF) at the BS. Thereby removing the need to feedback channel gains to the BS, PAF optimization, error propagation due to multiple SIC, and the need to send the SIC ordering and PAF information to the multiple downlink users. Moreover, since the proposed AE frameworks remove the SIC thus, we can decode the superposed signal of two users on the same SNR, removing the necessity of users to have a minimum of 3 – 8 dB of SNR gap for the successful implementation of SIC. Apart from achieving these benefits, for the AE-based BCM and d-BCM design, we show that we can decode all the multiplexed users' signals using the proposed AE frameworks even without the CSI knowledge at the receivers, which remains mandatory to implement SIC in conventional NOMA. Furthermore, as the number of downlink users increases the BER performance gains also increases. This shows the

capabilities of the AE framework in handling a large number of small IoT devices in future networks.

The AE framework's NN decoders give soft probabilistic outputs which we have shown to correspond to log-likelihood ratios (LLRs). Thus for the AE-based designed inner shortcodes, we utilize 5G-NR low-density parity-check (LDPC) codes with base graph 2 (BG2) and rate $1/5$ as outer codes. In Chapter 8, we note that the BER performance gains achieved by the AE-based BCM and d-BCM designs over the MLDs for short block lengths are even improved significantly by using the outer LDPC codes.

10.1.3 Special Notes Vision AE frameworks

In Chapter 9, we proposed an ML-based PL prediction model for urban street canyon using the 28 GHz measurement data collected from Manhattan. The feature set contains street clutter obtained from LiDAR point cloud and buildings from 3D mesh-grid. To mitigate the risk of overfitting, we defined seven expert features with physical meaning from the point cloud. We also compressed 3D building features to a length-12 vector for each link using convolutional NN-based AE. Instead of interpolation to nearby links, we focused on the extrapolation by introducing a street-by-street training and testing approach. Using linear ML algorithms for PL prediction, we achieved the superior PL prediction capability of our model in extrapolation. An intuitive interpretation of feature importance was also obtained using Lasso regression-based analysis and feature-exclusion analysis.

10.2 Future Works

In this section, we detail the future works of the proposed thesis.

In our recent collaborative work [161] with NOKIA Bell Labs, we designed an ultra-long-haul transmission of a new four-dimensional (4D) modulation format using AE frameworks. My contributions were specifically in the designing of AE-based BCM designs for the optical network while including the non-linear constraints in the designed framework³. The AE framework was practically demonstrated to significantly improve transmission distance and low generalized MI gap to Shannon limits. We proposed the AE framework in this thesis for wireless communication networks. We aimed to extend this work on a practical testbed developed in our lab.

Further, the AE works have recently been extended for MIMO communication networks [16] under fading channels for AE-based modulation-demodulation designs. However, AE-based BCM and d-BCM designs for MIMO networks still need to be investigated. Further, in Chapter 8, we proposed AE-based BCM and d-BCM designs for multi-

³Please note that due to the business secrets, we can not disclose the AE framework; thus, this work is not included in the thesis.

user downlink networks. Finally, in Chapter 9, we investigated ML-based PL prediction for mm-wave propagation in an urban street canyon with real-world measurements. In the future, we aim to consider Manhattan Street canyons with real-world mm-wave propagation for multiple downlink NOMA users. Specifically, we aim to extend this thesis by proposing an AE-based framework for the BCM and d-BCM for the downlink multi-user MIMO networks operating at the mm-wave frequencies.

We also aim to address the upcoming challenges of re-training the AE framework in different propagation conditions, using techniques like transfer learning [162].

Bibliography

- [1] Proakis, *Digital Communications 5th Edition*. McGraw Hill, 2007.
- [2] A. Goldsmith, "Joint source/channel coding for wireless channels," in *1995 IEEE 45th Vehicular Technology Conference. Countdown to the Wireless Twenty-First Century*, vol. 2, 1995, pp. 614–618 vol.2.
- [3] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 873–884, 1992.
- [4] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 55–67, 1982.
- [5] S.-C. Ma and M.-C. Lin, "A trellis coded modulation scheme constructed from block coded modulation with interblock memory," *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1348–1363, 1994.
- [6] T. F. T. Kasami, T. Takata and S. Lir, "Error control systems with combined block coding and m-ary PSK modulations," *Proceedings 10th Symposium Information Theory and its Applications*, Enoshima Island, Japan, 1987.
- [7] R. G. C. Williams, "Low Complexity Block Coded Modulation," Ph.D. dissertation, Computer engineering, The University of Manchester (United Kingdom), 1988.
- [8] B. Vucetic and S. Lin, "Block coded modulation and concatenated coding schemes for error control on fading channels," *Discrete Applied Mathematics*, vol. 33, no. 1, pp. 257–269, 1991. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0166218X9190121C>
- [9] H. de Oliveira and G. Battail, "The Random Coded Modulation: Performance And Euclidean Distance Spectrum Evaluation," in *Proceedings. 1991 IEEE International Symposium on Information Theory*, 1991, pp. 2–2.

- [10] S. Ali, W. Saad, N. Rajatheva, K. Chang, D. Steinbach, B. Sliwa, C. Wietfeld, K. Mei, H. Shiri, H.-J. Zepernick, T. M. C. Chu, I. Ahmad, J. Huusko, J. Suutala, S. Bhadauria, V. Bhatia, R. Mitra, S. Amuru, R. Abbas, B. Shao, M. Capobianco, G. Yu, M. Claes, T. Karvonen, M. Chen, M. Girnyk, and H. Malik, "6G White Paper on Machine Learning in Wireless Communication Networks," *arXiv*, *arXiv:2004.13875*, 2020.
- [11] H. Viswanathan and P. E. Mogensen, "Communications in the 6G Era," *IEEE Access*, vol. 8, pp. 57 063–57 074, 2020.
- [12] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.
- [13] T. Schenk, *RF Imperfections in High-Rate Wireless Systems: Impact and Digital Compensation*, 1st ed. Springer Publishing Company, Incorporated, 2008.
- [14] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0893608089900208>
- [15] E. Balevi and J. G. Andrews, "Autoencoder-Based Error Correction Coding for One-Bit Quantization," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3440–3451, 2020.
- [16] Q. Hu, F. Gao, H. Zhang, G. Y. Li, and Z. Xu, "Understanding Deep MIMO Detection," *arXiv*, *arXiv:2105.05044*, 2021.
- [17] Y. Zhao, R. Adve, and T. J. Lim, "Improving Amplify-and-Forward Relay Networks: Optimal Power Allocation versus Selection," in *2006 IEEE International Symposium on Information Theory*, 2006, pp. 1234–1238.
- [18] W. Su, A. K. Sadek, and K. J. Ray Liu, "Cooperative Communication Protocols in Wireless Networks: Performance Analysis and Optimum Power Allocation," *Wireless Personal Communications*, vol. 44, no. 2, pp. 181–217, 2008. [Online]. Available: <https://doi.org/10.1007/s11277-007-9359-z>
- [19] A. E. Canbilen, S. S. Ikki, E. Basar, S. S. Gultekin, and I. Develi, "Impact of I/Q Imbalance on Amplify-and-Forward Relaying: Optimal Detector Design and Error Performance," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3154–3166, 2019.

- [20] K. Singh, A. Gupta, and T. Ratnarajah, “QoS-Driven Resource Allocation and EE-Balancing for Multiuser Two-Way Amplify-and-Forward Relay Networks,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3189–3204, 2017.
- [21] ———, “Green resource allocation and EE-balancing in multiuser two-way amplify-and-forward relay networks,” in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
- [22] ———, “A Utility-Based Joint Subcarrier and Power Allocation for Green Communications in Multi-User Two-Way Regenerative Relay Networks,” *IEEE Transactions on Communications*, vol. 65, no. 9, pp. 3705–3722, 2017.
- [23] K. Singh, A. Gupta, M.-L. Ku, and T. Ratnarajah, “Joint Subcarrier Pairing and Power Allocation for Two-Way Energy-Efficient Relay Networks,” in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.
- [24] A. Gupta, S. Biswas, K. Singh, T. Ratnarajah, and M. Sellathurai, “An Energy-Efficient Approach Towards Power Allocation in Non-Orthogonal Multiple Access Full-Duplex AF Relay Systems,” in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.
- [25] A. Gupta, K. Singh, and M. Sellathurai, “Time-Switching EH-Based Joint Relay Selection and Resource Allocation Algorithms for Multi-User Multi-Carrier AF Relay Networks,” *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 2, pp. 505–522, 2019.
- [26] G. Böcherer, *Principles of coded modulation*. Technische Universität München, 2018, <http://www.georg-boecherer.de/bocherer2018principles.pdf>.
- [27] S. Cammerer, F. A. Aoudia, S. Dörner, M. Stark, J. Hoydis, and S. ten Brink, “Trainable Communication Systems: Concepts and Prototype,” *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5489–5503, 2020.
- [28] M. Sadeghi and E. G. Larsson, “Physical Adversarial Attacks Against End-to-End Autoencoder Communication Systems,” *IEEE Communications Letters*, vol. 23, no. 5, pp. 847–850, 2019.
- [29] X. Chen, J. Cheng, Z. Zhang, L. Wu, J. Dang, and J. Wang, “Data-Rate Driven Transmission Strategies for Deep Learning-Based Communication Systems,” *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2129–2142, 2020.
- [30] M. E. Morocho-Cayamcela and W. Lim, “Accelerating wireless channel autoencoders for short coherence-time communications,” *Journal of Communications and Networks*, vol. 22, no. 3, pp. 215–222, 2020.

- [31] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, “Deep Learning Based Communication Over the Air,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2018.
- [32] N. Wu, X. Wang, B. Lin, and K. Zhang, “A CNN-Based End-to-End Learning Framework Toward Intelligent Communication Systems,” *IEEE Access*, vol. 7, pp. 110 197–110 204, 2019.
- [33] V. Raj and S. Kalyani, “Design of Communication Systems Using Deep Learning: A Variational Inference Perspective,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1320–1334, 2020.
- [34] F. A. Aoudia and J. Hoydis, “Model-Free Training of End-to-End Communication Systems,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, 2019.
- [35] D. Wu, M. Nekovee, and Y. Wang, “Deep Learning-Based Autoencoder for m-User Wireless Interference Channel Physical Layer Design,” *IEEE Access*, vol. 8, pp. 174 679–174 691, 2020.
- [36] Y. Lu, P. Cheng, Z. Chen, Y. Li, W. H. Mow, and B. Vucetic, “Deep Autoencoder Learning for Relay-Assisted Cooperative Communication Systems,” *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5471–5488, 2020.
- [37] Y. Lu, P. Cheng, Z. Chen, W. H. Mow, and Y. Li, “A Learning Approach to Cooperative Communication System Design,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5240–5244.
- [38] T. Matsumine, T. Koike-Akino, and Y. Wang, “Deep Learning-Based Constellation Optimization for Physical Network Coding in Two-Way Relay Networks,” in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.
- [39] Y. Lu, P. Cheng, Z. Chen, W. H. Mow, Y. Li, and B. Vucetic, “Deep Multi-Task Learning for Cooperative NOMA: System Design and Principles,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 61–78, 2021.
- [40] T. V. Luong, Y. Ko, M. Matthaiou, N. A. Vien, M.-T. Le, and V.-D. Ngo, “Deep Learning-Aided Multicarrier Systems,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 2109–2119, 2021.
- [41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.

- [42] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv, arXiv:1412.6980*, 2017.
- [43] N. Ketkar, “Introduction to keras,” in *Deep Learning with Python*. Springer, 2017, pp. 97–111.
- [44] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [45] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterton, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: <http://proceedings.mlr.press/v9/glorot10a.html>
- [46] M. Li, M. Soltanolkotabi, and S. Oymak, “Gradient Descent with Early Stopping is Provably Robust to Label Noise for Overparameterized Neural Networks,” *arXiv, arXiv:1903.11680*, 2019.
- [47] M. Shirvanimoghaddam, M. S. Mohammadi, R. Abbas, A. Minja, C. Yue, B. Matuz, G. Han, Z. Lin, W. Liu, Y. Li, S. Johnson, and B. Vucetic, “Short Block-Length Codes for Ultra-Reliable Low Latency Communications,” *IEEE Communications Magazine*, vol. 57, no. 2, pp. 130–137, 2019.
- [48] Y. Liu, P. M. Olmos, and D. G. M. Mitchell, “Generalized LDPC Codes for Ultra Reliable Low Latency Communication in 5G and Beyond,” *IEEE Access*, vol. 6, pp. 72 002–72 014, 2018.
- [49] *Introduction and Preview, Elements of Information Theory*. John Wiley Sons, Ltd, 2005, ch. 1-17, pp. 1–687. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/047174882X.ch1>
- [50] M. M. G. B. A. A. Choromanska, M. Henaff and Y. L. Cun, “The Loss surfaces of Multilayer Networks,” in *18th Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2015, p. 192–204.

- [51] C. G. K. C. S. G. Y. N. Dauphin, R. Pascanu and Y. Bengio, “Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-convex Optimization,” in *27th Advances in Neural Information Processing Systems*, 2014, p. 2933–2941.
- [52] D. Masters and C. Luschi, “Revisiting Small Batch Training for Deep Neural Networks,” *arXiv, arXiv:1804.07612*, 2018.
- [53] Y. Huang, Y. Cheng, A. Bapna, O. Firat, M. X. Chen, D. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and Z. Chen, “GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism,” *arXiv, arXiv:1811.06965*, 2019.
- [54] L. van der Maaten and G. Hinton, “Visualizing Data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: <http://jmlr.org/papers/v9/vandermaaten08a.html>
- [55] E. Agrell, “Database of sphere packings.” [Online]. Available: <https://codes.se/packings/>
- [56] K. Singh, A. Gupta, T. Ratnarajah, and M.-L. Ku, “A General Approach Toward Green Resource Allocation in Relay-Assisted Multiuser Communication Networks,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 848–862, 2018.
- [57] A. Gupta and M. Sellathurai, “End-to-End Learning-based Amplify-and-Forward Relay Networks using Autoencoders,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.
- [58] M. M. Harb and M. F. Al-Mistarihi, “Dual Hop Differential Amplify-and-Forward relaying with selection combining cooperative diversity over Nakagami-m fading channels,” in *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, 2016, pp. 225–228.
- [59] M. R. Aveni and H. H. Nguyen, “Selection Combining for Differential Amplify-and-Forward Relaying Over Rayleigh-Fading Channels,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 277–280, 2013.
- [60] Z. Fang, L. Zheng, L. Wang, and L. Jin, “A frequency domain differential modulation scheme for asynchronous amplify-and-forward relay networks,” in *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, 2015, pp. 977–981.
- [61] T. Himsoon, W. Su, and K. R. Liu, “Differential transmission for amplify-and-forward Cooperative communications,” *IEEE Signal Processing Letters*, vol. 12, no. 9, pp. 597–600, 2005.

- [62] M. R. Aveni and H. H. Nguyen, "Performance of Selection Combining for Differential Amplify-and-Forward Relaying Over Time-Varying Channels," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4156–4166, 2014.
- [63] Y. Lou, Y.-K. Ma, Q.-Y. Yu, H.-L. Zhao, and W. Xiang, "A Differential ML Combiner for Differential Amplify-and-Forward System in Time-Selective Fading Channels," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 10 157–10 163, 2016.
- [64] W. Cho, R. Cao, and L. Yang, "Optimum Resource Allocation for Amplify-and-Forward Relay Networks With Differential Modulation," *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5680–5691, 2008.
- [65] T. Koike-Akino, P. Popovski, and V. Tarokh, "Optimized constellations for two-way wireless relaying with physical network coding," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 773–787, 2009.
- [66] T. T. Nguyen and L. Lampe, "Bit-Interleaved Coded Modulation with Mismatched Decoding Metrics," *IEEE Transactions on Communications*, vol. 59, no. 2, pp. 437–447, 2011.
- [67] A. Gupta and M. Sellathurai, "A Stacked-Autoencoder Based End-to-End Learning Framework for Decode-and-Forward Relay Networks," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5245–5249.
- [68] J. Li, M. Matthaiou, and T. Svensson, "I/Q Imbalance in AF Dual-Hop Relaying: Performance Analysis in Nakagami-m Fading," *IEEE Transactions on Communications*, vol. 62, no. 3, pp. 836–847, 2014.
- [69] J. Qi, S. Aïssa, and M.-S. Alouini, "Impact of I/Q imbalance on the performance of two-way CSI-assisted AF relaying," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, 2013, pp. 2507–2512.
- [70] E. Bjornson, M. Matthaiou, and M. Debbah, "A New Look at Dual-Hop Relaying: Performance Limits with Hardware Impairments," *IEEE Transactions on Communications*, vol. 61, no. 11, pp. 4512–4525, 2013.
- [71] Y. Gao, Y. Chen, N. Chen, and J. Zhang, "Performance Analysis of Dual-Hop Relaying With I/Q Imbalance and Additive Hardware Impairment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4580–4584, 2020.

- [72] W. Hou and M. Jiang, "Enhanced Joint Channel and IQ Imbalance Parameter Estimation for Mobile Communications," *IEEE Communications Letters*, vol. 17, no. 7, pp. 1392–1395, 2013.
- [73] A. Gupta and M. Sellathurai, "End-to-End Learning-Based Framework for Amplify-and-Forward Relay Networks," *IEEE Access*, vol. 9, pp. 81 660–81 677, 2021.
- [74] G. Liva, L. Gaudio, T. Ninacs, and T. Jerkovits, "Code Design for Short Blocks: A Survey," *arXiv, arXiv:1610.00873*, 2016.
- [75] 3GPP, "5G; NR; Multiplexing and channel coding," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.212, 01 2020, version 15.8.0. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.08.00_60/ts_138212v150800p.pdf
- [76] Y. Zhang, K. Peng, X. Wang, and J. Song, "Performance Analysis and Code Optimization of IDMA With 5G New Radio LDPC Code," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1552–1555, 2018.
- [77] X. Wu, M. Jiang, C. Zhao, L. Ma, and Y. Wei, "Low-Rate PBRL-LDPC Codes for URLLC in 5G," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 800–803, 2018.
- [78] K. Singh, A. Gupta, and T. Ratnarajah, "QoS-Driven Energy-Efficient Resource Allocation in Multiuser Amplify-and-Forward Relay Networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 4, pp. 771–786, 2017.
- [79] ———, "Energy Efficient Resource Allocation for Multiuser Relay Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 1218–1235, 2017.
- [80] ———, "Efficient joint subcarrier and power allocation for achieving green multiuser full-duplex decode-and-forward relay networks," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
- [81] K. Singh, A. Gupta, S. Biswas, and T. Ratnarajah, "A Unified Approach Towards Green Resource Allocation in Relay-Assisted Multiuser Networks," in *GLOBE-COM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.
- [82] J. Park, D. J. Ji, and D.-H. Cho, "High-Order Modulation Based on Deep Neural Network for Physical-Layer Network Coding," *IEEE Wireless Communications Letters*, vol. 10, no. 6, pp. 1173–1177, 2021.

- [83] A. Hirose, *Complex-Valued Neural Networks: Theories and Applications (Series on Innovative Intelligence, 5)*. World Scientific Press, 2004.
- [84] L. Han, C. Huang, S. Shao, and Y. Tang, "Relay Placement for Amplify-and-Forward Relay Channels with Correlated Shadowing," *IEEE Wireless Communications Letters*, vol. 2, no. 2, pp. 171–174, 2013.
- [85] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. USA: Cambridge University Press, 2005.
- [86] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, "Practical, Real-Time, Full Duplex Wireless," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 301–312. [Online]. Available: <https://doi.org/10.1145/2030613.2030647>
- [87] Z. Zhang, K. Long, A. V. Vasilakos, and L. Hanzo, "Full-Duplex Wireless Communications: Challenges, Solutions, and Future Research Directions," *Proceedings of the IEEE*, vol. 104, no. 7, pp. 1369–1409, 2016.
- [88] J. Xue, S. Biswas, A. C. Cirik, H. Du, Y. Yang, T. Ratnarajah, and M. Sellathurai, "Transceiver Design of Optimum Wirelessly Powered Full-Duplex MIMO IoT Devices," *IEEE Transactions on Communications*, vol. 66, no. 5, pp. 1955–1969, 2018.
- [89] C. Zhong, M. Matthaiou, G. K. Karagiannidis, and T. Ratnarajah, "Generic Ergodic Capacity Bounds for Fixed-Gain AF Dual-Hop Relaying Systems," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 8, pp. 3814–3824, 2011.
- [90] Z. Ding, T. Ratnarajah, and K. K. Leung, "On the Study of Network Coded AF Transmission Protocol for Wireless Multiple Access Channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4568–4574, 2008.
- [91] A. Bishnu, M. Holm, and T. Ratnarajah, "Performance Evaluation of Full-Duplex IAB Multi-Cell and Multi-User Network for FR2 Band," *IEEE Access*, vol. 9, pp. 72 269–72 283, 2021.
- [92] H. He, S. Biswas, P. Aquilina, T. Ratnarajah, and J. Yang, "Performance Analysis of Multi-Cell Full-Duplex Cellular Networks," *IEEE Access*, vol. 8, pp. 206 914–206 930, 2020.
- [93] H. Guo, S. Wu, H. Wang, and M. Daneshmand, "DSIC: Deep Learning Based Self-Interference Cancellation for In-Band Full Duplex Wireless," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

- [94] J. R. Krier and I. F. Akyildiz, "Active self-interference cancellation of passband signals using gradient descent," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013, pp. 1212–1216.
- [95] T. P. Do and T. V. T. Le, "Power Allocation and Performance Comparison of Full Duplex Dual Hop Relaying Protocols," *IEEE Communications Letters*, vol. 19, no. 5, pp. 791–794, 2015.
- [96] K.-G. Wu, F.-T. Chien, Y.-F. Lin, and M.-K. Chang, "SINR and Delay Analyses in Two-Way Full-Duplex SWIPT-Enabled Relaying Systems," *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2148–2162, 2021.
- [97] K. Yang, H. Cui, L. Song, and Y. Li, "Efficient Full-Duplex Relaying With Joint Antenna-Relay Selection and Self-Interference Suppression," *IEEE Transactions on Wireless Communications*, vol. 14, no. 7, pp. 3991–4005, 2015.
- [98] L. Jiménez Rodríguez, N. H. Tran, and T. Le-Ngoc, "Performance of Full-Duplex AF Relaying in the Presence of Residual Self-Interference," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 9, pp. 1752–1764, 2014.
- [99] T. K. Baranwal, D. S. Michalopoulos, and R. Schober, "Outage Analysis of Multihop Full Duplex Relaying," *IEEE Communications Letters*, vol. 17, no. 1, pp. 63–66, 2013.
- [100] S.-J. Park and M.-K. Byeon, "Irregularly distributed triangular quadrature amplitude modulation," in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2008, pp. 1–5.
- [101] T. G. Markiewicz, "Construction and Labeling of Triangular QAM," *IEEE Communications Letters*, vol. 21, no. 8, pp. 1751–1754, 2017.
- [102] A. R. Heidarpour, M. Ardakani, and C. Tellambura, "Network Coded Cooperation Based on Relay Selection with Imperfect CSI," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–5.
- [103] M. Chen and A. Yener, "Power allocation for F/TDMA multiuser two-way relay networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 2, pp. 546–551, 2010.
- [104] S. J. Kim, P. Mitran, and V. Tarokh, "Performance Bounds for Bidirectional Coded Cooperation Protocols," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 5235–5241, 2008.

- [105] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A Survey on Non-Orthogonal Multiple Access for 5G Networks: Research Challenges and Future Trends," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2181–2195, 2017.
- [106] S. M. R. Islam, N. Avazov, O. A. Dobre, and K.-s. Kwak, "Power-Domain Non-Orthogonal Multiple Access (NOMA) in 5G Systems: Potentials and Challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 721–742, 2017.
- [107] T. Yazaki and Y. Sanada, "Effect of joint detection and decoding in non-orthogonal multiple access," in *2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2014, pp. 245–250.
- [108] C.-H. Lin, S.-L. Shieh, T.-C. Chi, and P.-N. Chen, "Optimal Inter-Constellation Rotation Based on Minimum Distance Criterion for Uplink NOMA," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 525–539, 2019.
- [109] B. Di, L. Song, Y. Li, and G. Y. Li, "TCM-NOMA: Joint Multi-User Codeword Design and Detection in Trellis-Coded Modulation-Based NOMA for Beyond 5G," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 766–780, 2019.
- [110] W. Wang and W. Zhang, "Optimal signal constellation for downlink two-user NOMA," in *2017 23rd Asia-Pacific Conference on Communications (APCC)*, 2017, pp. 1–6.
- [111] Y. Zhao, J. Hu, Z. Ding, and K. Yang, "Constellation Rotation Aided Modulation Design for the Multi-User SWIPT-NOMA," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.
- [112] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep Learning for an Effective Nonorthogonal Multiple Access Scheme," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8440–8450, 2018.
- [113] M. Kim, N.-I. Kim, W. Lee, and D.-H. Cho, "Deep Learning-Aided SCMA," *IEEE Communications Letters*, vol. 22, no. 4, pp. 720–723, 2018.
- [114] N. Ye, X. Li, H. Yu, A. Wang, W. Liu, and X. Hou, "Deep Learning Aided Grant-Free NOMA Toward Reliable Low-Latency Access in Tactile Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2995–3005, 2019.
- [115] F. Sun, K. Niu, and C. Dong, "Deep Learning Based Joint Detection and Decoding of Non-Orthogonal Multiple Access Systems," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–5.

- [116] F. Alberge, “Constellation design with deep learning for downlink non-orthogonal multiple access,” in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 1–5.
- [117] E. Stauffer, A. Wang, and N. Jindal, “Deep Learning for the Degraded Broadcast Channel,” in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 1760–1763.
- [118] L. Jiang, X. Li, N. Ye, and A. Wang, “Deep Learning-Aided Constellation Design for Downlink NOMA,” in *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, 2019, pp. 1879–1883.
- [119] T. Assaf, A. Al-Dweik, M. E. Moursi, and H. Zeineldin, “Exact BER Performance Analysis for Downlink NOMA Systems Over Nakagami-m Fading Channels,” *IEEE Access*, vol. 7, pp. 134 539–134 555, 2019.
- [120] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *arXiv, arXiv:1502.03167*, 2015.
- [121] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.
- [122] C. Phillips, D. Sicker, and D. Grunwald, “A Survey of Wireless Path Loss Prediction and Coverage Mapping Methods,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 255–270, 2013.
- [123] V. Erceg, “Channel Models for Broadband Wireless Access,” IEEE 802.16 Broadband Wireless Access Working Group, For information and discussion Session 10, 11 2007, v.3c. [Online]. Available: https://www.ieee802.org/16/tg3/contrib/802163p-00_47.pdf
- [124] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, “Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!” *IEEE Access*, vol. 1, pp. 335–349, 2013.
- [125] T. S. Rappaport, G. R. MacCartney, S. Sun, H. Yan, and S. Deng, “Small-Scale, Local Area, and Transitional Millimeter Wave Propagation for 5G Communications,” *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, pp. 6474–6490, 2017.
- [126] J. Du, D. Chizhik, R. Feick, M. Rodríguez, G. Castro, and R. A. Valenzuela, “Suburban Fixed Wireless Access Channel Measurements and Models at 28 GHz for 90% Outdoor Coverage,” *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 1, pp. 411–420, 2020.

- [127] “5G Channel Model for bands up to 100 GHz, Annex A: Summary of channel sounding, simulations and measurement data,” 5GCM White Paper, 03 2016. [Online]. Available: <http://www.5gworkshops.com/5GCM.html>
- [128] J. Du, D. Chizhik, R. A. Valenzuela, R. Feick, G. Castro, M. Rodriguez, T. Chen, M. Kohli, and G. Zussman, “Directional Measurements in Urban Street Canyons From Macro Rooftop Sites at 28 GHz for 90% Outdoor Coverage,” *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 6, pp. 3459–3469, 2021.
- [129] T. Sarkar, Z. Ji, K. Kim, A. Medouri, and M. Salazar-Palma, “A survey of various propagation models for mobile communication,” *IEEE Antennas and Propagation Magazine*, vol. 45, no. 3, pp. 51–82, 2003.
- [130] E. Montiel, A. Aguado, and F. Sillion, “A radiance model for predicting radio wave propagation in irregular dense urban areas,” *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 11, pp. 3097–3108, 2003.
- [131] I. Popescu, D. Nikitopoulos, P. Constantinou, and I. Nafornita, “Comparison of ANN Based Models for Path Loss Prediction in Indoor Environment,” in *IEEE Vehicular Technology Conference*, 2006, pp. 1–5.
- [132] E. Ostlin, H.-J. Zepernick, and H. Suzuki, “Macrocell Path-Loss Prediction Using Artificial Neural Networks,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2735–2747, 2010.
- [133] M. Ribero, R. W. Heath, H. Vikalo, D. Chizhik, and R. A. Valenzuela, “Deep Learning Propagation Models over Irregular Terrain,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 4519–4523.
- [134] D. Wu, G. Zhu, and B. Ai, “Application of artificial neural networks for path loss prediction in railway environments,” in *2010 5th International ICST Conference on Communications and Networking in China*, 2010, pp. 1–5.
- [135] S. P. Sotiroudis, S. K. Goudos, and K. Siakavara, “Neural Networks and Random Forests: A Comparison Regarding Prediction of Propagation Path Loss for NB-IoT Networks,” in *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCASST)*, 2019, pp. 1–4.
- [136] H. F. Ates, S. M. Hashir, T. Baykas, and B. K. Gunturk, “Path Loss Exponent and Shadowing Factor Prediction From Satellite Images Using Deep Learning,” *IEEE Access*, vol. 7, pp. 101 366–101 375, 2019.

- [137] Y. Egi and C. E. Otero, "Machine-Learning and 3D Point-Cloud Based Signal Power Path Loss Model for the Deployment of Wireless Communication Systems," *IEEE Access*, vol. 7, pp. 42 507–42 517, 2019.
- [138] U. Masood, H. Farooq, and A. Imran, "A Machine Learning Based 3D Propagation Model for Intelligent Future Cellular Networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.
- [139] J. Isabona and V. M. Srivastava, "Hybrid neural network approach for predicting signal propagation loss in urban microcells," in *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2016, pp. 1–5.
- [140] C. Park, D. K. Tettey, and H.-S. Jo, "Artificial Neural Network Modeling for Path Loss Prediction in Urban Environments," *arXiv*, *arXiv:1904.02383*, 2019.
- [141] Y. Zhang, J. Wen, G. Yang, Z. He, and J. Wang, "Path Loss Prediction Based on Machine Learning: Principle, Method, and Data Expansion," *Applied Sciences*, vol. 9, no. 9, 2019. [Online]. Available: <https://www.mdpi.com/2076-3417/9/9/1908>
- [142] M. E. Morocho-Cayamcela, M. Maier, and W. Lim, "Breaking Wireless Propagation Environmental Uncertainty With Deep Learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5075–5087, 2020.
- [143] C. A. Oroza, Z. Zhang, T. Watteyne, and S. D. Glaser, "A Machine-Learning-Based Connectivity Model for Complex Terrain Large-Scale Low-Power Wireless Deployments," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 576–584, 2017.
- [144] S. I. Popoola, E. Adetiba, A. A. Atayero, N. Faruk, and C. T. Calafate, "Optimal model for path loss predictions using feed-forward neural networks," *Cogent Engineering*, vol. 5, no. 1, p. 1444345, 2018. [Online]. Available: <https://doi.org/10.1080/23311916.2018.1444345>
- [145] D. C. J. D. Y. Wang, S. R. Iyer and R. A. Valenzuela, "Channel prediction with terrains: deep autoencoder with random forest," Bell Labs, Technical Report, 09 2019, iTD-19-59529F. [Online]. Available: https://www.ieee802.org/16/tg3/contrib/802163p-00_47.pdf
- [146] N. Kuno and Y. Takatori, "Prediction Method by Deep-Learning for Path Loss Characteristics in an Open-Square Environment," in *2018 International Symposium on Antennas and Propagation (ISAP)*, 2018, pp. 1–2.

- [147] J. Thrane, D. Zibar, and H. L. Christiansen, "Model-Aided Deep Learning Method for Path Loss Prediction in Mobile Communication Systems at 2.6 GHz," *IEEE Access*, vol. 8, pp. 7925–7936, 2020.
- [148] R. Levie, Yapar, G. Kutyniok, and G. Caire, "RadioUNet: Fast Radio Map Estimation With Convolutional Neural Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 4001–4015, 2021.
- [149] J.-Y. Lee, M. Y. Kang, and S.-C. Kim, "Path Loss Exponent Prediction for Outdoor Millimeter Wave Channels through Deep Learning," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–5.
- [150] V. V. Ratnam, H. Chen, S. Pawar, B. Zhang, C. J. Zhang, Y.-J. Kim, S. Lee, M. Cho, and S.-R. Yoon, "FadeNet: Deep Learning-Based mm-Wave Large-Scale Channel Fading Prediction and its Applications," *IEEE Access*, vol. 9, pp. 3278–3290, 2021.
- [151] A. Seretis and C. D. Sarris, "An Overview of Machine Learning Techniques for Radiowave Propagation Modeling," *arXiv*, *arXiv:2101.11760*, 2021.
- [152] "Dataset of 2013 - 2014 USGS CMGP LiDAR: Post Sandy (New York City)," United States Geological Survey (USGS), 10 2014. [Online]. Available: https://coast.noaa.gov/htdata/lidar1_z/geoid12b/data/4920/
- [153] "New York City Building CAD Files Dataset," CADMAPPER. [Online]. Available: <https://cadmapper.com/>
- [154] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.901, 05 2017, version 14.0.0. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/14.00.00_60/tr_138901v140000p.pdf
- [155] M. Marcus and B. Pattan, "Millimeter wave propagation: spectrum management implications," *IEEE Microwave Magazine*, vol. 6, no. 2, pp. 54–62, 2005.
- [156] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-Wave Cellular Wireless Networks: Potentials and Challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, 2014.
- [157] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D Point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008, semantic Knowledge in Robotics. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889008001140>

- [158] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: <http://jmlr.org/papers/v12/pedregosa11a.html>
- [159] M. Crane, “Questionable Answers in Question Answering Research: Reproducibility and Variability of Published Results,” *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 241–252, 2018. [Online]. Available: <https://aclanthology.org/Q18-1018>
- [160] M. N. W. G. B. H. Tomislav Hengl, Madlene Nussbaum and B. Gräler, “Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables,” *PeerJ*, vol. 6:e5518, 2018. [Online]. Available: <https://doi.org/10.7717/peerj.5518>
- [161] R.-J. Essiambre, R. Ryf, M. Kodialam, B. Chen, M. Mazur, J. I. Bonetti, R. Veronese, H. Huang, A. Gupta, F. A. Aoudia, E. C. Burrows, D. F. Grosz, L. Palmieri, M. Sellathurai, X. Chen, N. K. Fontaine, and H. Chen, “Increased Reach of Long-Haul Transmission using a Constant-Power 4D Format Designed Using Neural Networks,” in *2020 European Conference on Optical Communications (ECOC)*, 2020, pp. 1–4.
- [162] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A Survey on Deep Transfer Learning,” *arXiv*, *arXiv:1808.01974*, 2018.