Applied Deep Learning for Artificial Intelligence-enabled Wireless Communication

A Thesis submitted by

Christopher P. Davey

BA(VisArts Hons), MInfoTech, MSc (Mathematics and Statistics)

For the award of

Doctor of Philosophy

2024

ABSTRACT

Wireless communications systems are ubiquitous capabilities underlying the technologies which are transforming our society, business, government, and industry today. Ever evolving applications are driving the requirements for the design of such wireless communication systems. Demanding increased speed and volumes in data transfer, the reliability and availability, and important protections such as the privacy as well as the essential security of these communications systems. Future wireless communications systems must approach these design challenges by addressing these emerging requirements and look for opportunities to leverage emerging technologies to complement the conventional design methods. Exponential growth in computing hardware and processing capabilities have supported the application of machine learning, and in particular deep learning, to extract value from large scale data assets. Researchers have also recognised the potential for the application of deep learning to the data-driven design of wireless communications systems, complex channel environments and emerging applications. This doctoral research thesis develops artificial intelligence-enabled deep learning models and training algorithms to specifically focus on four key objectives in application to wireless communications systems design. These objectives are **Synchronisation** (Objective 1), **Adaptation** (Objective 2) and **Over-the-air learning** (Objectives 3 & 4). **Synchronisation** is an important signal processing step in the receiver that aims to correct the perturbed signals to retrieve the original message. This thesis proposes a new method of parameter estimation for **Synchronisation** supporting multiple modulations, including chaotic modulations for secure communications. This objective aims to maximise the data-rate and security of wireless communications by focusing on short random preamble sequences which severely limit the accuracy of conventional methods. **Adaptation** is a desirable property supporting the reliability and availability of wireless communications systems in changing channel conditions. In this objective, a custom deep learning architecture is developed under a multi-task learning framework to learn multiple code-rates and is demonstrated to produce gains under fading channel environments in comparison to conventional coding methods. **Over-the-air learning**, as part of objectives 3 & 4, makes a novel contribution in the area of adaptation while enabling the global optimisation of both transmitter and receiver for a true channel environment. This objective addresses the challenging tasks of complex training and modelling regimes found in rapidly evolving wireless communication systems. Therefore, the thesis proposes two novel methods simplifying the training procedure and deep learning models for **Over-the-air learning** and addresses the reliability and availability in changing channel conditions as well as security during the training procedure. The newly proposed methods in this doctoral thesis clearly demonstrate the success of the proposed approaches for simulation, generalisation, training techniques and custom deep learning architectures. The research project outcomes are useful for establishing practical pathways for future applications of artificial intelligence-enabled wireless communications systems.

Certification of Thesis

(All candidates to reproduce this section in their thesis verbatim)

I Christopher Davey declare that the PhD Thesis entitled *Applied Deep Learning for Artificial Intelligence-enabled Wireless Communication* is not more than 100,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references, and footnotes. This Thesis is the work of Christopher Davey except where otherwise acknowledged, with the majority of the contribution to the papers presented as a Thesis by Publication undertaken by the student. The work is original and has not previously been submitted for any other award, except where acknowledged.

Date: Tuesday 21st May, 2024

Endorsed by:

Professor Ravinesh C. Deo
Principal Supervisor

Adjunct Professor Ismail Shakeel
Associate Supervisor

Professor Jeffrey Soar
Associate Supervisor

Professor Sancho Salcedo-Sanz
Associate Supervisor

Student and supervisors' signatures of endorsement are held at the University.

Statement of Contribution

This PhD thesis by publications has produced 4 high quality (Quartile 1) publications during the candidature. The details of joint authorship and agreed share of these contributions are detailed as follows:

Article 1: Chapter 4

Davey, C. P., Shakeel, I., Deo, R. C., Salcedo-Sanz, S., & Soar, J. (2022). Using Sequence-to-Sequence Models for Carrier Frequency Offset Estimation of Short Messages and Chaotic Maps. *IEEE Access*, 10, 119814-119825. doi:10.1109/ACCESS.2022.3221762
`https://ieeexplore.ieee.org/document/9947074`

Christopher Davey contributed 75% to this paper. Collectively Ismail Shakeel, Ravinesh C. Deo, Sancho Salcedo-Sanz and Jeffrey Soar contributed the remainder.

Article 2: Chapter 5

Davey, C. P., Shakeel, I., Deo, R. C., Sharma, E., Salcedo-Sanz, S., & Soar, J. (2024). End-to-End Learning of Adaptive Coded Modulation Schemes for Resilient Wireless Communications. *Applied Soft Computing*, 159, 111672, ISSN 1568-4946. doi:10.1016/j.asoc.2024.111672
`https://doi.org/10.1016/j.asoc.2024.111672`

Christopher Davey contributed 70% to this paper. Collectively Ismail Shakeel, Ravinesh C. Deo, Ekta Sharma, Sancho Salcedo-Sanz and Jeffrey Soar contributed the remainder.

Article 3: Chapter 6

Davey, C. P., Shakeel, I., Deo, R. C., & Salcedo-Sanz, S. (2023). Channel Agnostic Training of Transmitter and Receiver for Wireless Communications. *Sensors*, 23(24), ISSN 1424-8220. doi:10.3390/s23249848
`https://www.mdpi.com/1424-8220/23/24/9848`

Christopher Davey contributed 75% to this paper. Collectively Ismail Shakeel, Ravinesh C. Deo and Sancho Salcedo-Sanz contributed the remainder.

Article 4: Chapter 7

Davey, C. P., Shakeel, I., Deo, R. C., & Salcedo-Sanz, S. (2024). Deep Learning Based Over-the-Air Training of Wireless Communication Systems Without Feedback. *Sensors*, 24(10), ISSN 1424-8220. doi: 10.3390/s24102993
`https://www.mdpi.com/1424-8220/24/10/2993`

Christopher Davey contributed 75% to this paper. Collectively Ismail Shakeel, Ravinesh C. Deo and Sancho Salcedo-Sanz contributed the remainder.

ACKNOWLEDGMENTS

I would like to acknowledge the support of my supervisory team in providing guidance and collaboration towards the completion of this PhD program of research.

Firstly, I would like to thank my principal supervisor, Professor Ravinesh C. Deo, whose experience, enthusiasm and encouragement has buoyed my progress. His approach to publication and review processes has demonstrated to me tact, optimism and thoroughness that I have sought to emulate in responses to review throughout the publication process of this research journey. To look at critical review with fresh eyes, to interpret the sometime cryptic reviewer comments, and to think of how to improve given each response, and to evaluate what extent of change is necessary, has given me an approach to working with feedback that extends well beyond publication. Additionally, Ravinesh has helped me improve the structure of my writing a great deal, and I am very thankful.

Secondly, I would like to express my gratitude for my associate supervisor, Adjunct Professor Ismail Shakeel, whose expertise in the field of wireless communications systems engineering and machine learning is instrumental in the development of this PhD research. His generosity of time, enthusiasm and expertise on the topic and vision has greatly assisted in generating possible avenues for investigation throughout the cycle of this PhD program.

Thirdly, I would like to thank associate supervisor, Professor Jeffrey Soar, for his editorial and feedback during the early stages of my PhD, the proposal and first few articles have benefitted and were improved with his advice.

Fourth, I would like to thank my associate supervisor, Professor Sancho Salcedo-Sanz, who has provided excellent editorial support for the publications contained within this thesis. And has participated in several discussions on possible technical approaches to different research problems. His expertise and generous ideas on this topic have also assisted with the development of this work.

I would also like to thank my colleague, Dr Ekta Sharma, who has provided editorial support during this PhD as well as had frequent discussions where I have had the opportunity to share ideas on the topic.

As part of this program I have had the opportunity to work with DSTG on a practical cyber security project, and had the good fortune of meeting Dr Alex Chambers, Alexander Rohl and Matthew Roddha, all of which are engaged with a fascinating area of work, and I enjoyed working with them on our small project immensely. Alex is also working on his own PhD in the cyber security field, and I wish him well in that endeavour.

In addition, I'd like to acknowledge my cousin Mel, who as a non-technical person, and who provided a readership of one, checked for the readability of my publications, and provided feedback on where I could improve them.

Finally, I would like to express my gratitude to my wife, Julie, for her patience and willingness to listen to me explain the ideas I was working on. Our afternoon walks with our dog, Iggy, provided ample opportunity to disconnect from the work and turn our focus to anything else. The art of small talk is an important and underrated skill.

Funding Acknowledgement

This research was supported by DSP Scholarship (Project-Based) Agreement 10254/UniSQ-DSTG Postgraduate Research Scholarship (2021-2024) on ’Design of Efficient Artificial Intelligence Algorithms for Future Communication Systems’. The Scholar acknowledges the Department of Defence and the Commonwealth of Australia in its participation and contribution to the Research Project and all publications or presentations made by the Scholar during the term of this Agreement. The Scholar also acknowledges the Chief Defence Scientist Fellowship held by the DSTG Supervisor under a DSP Scholarship (Project-Based) Agreement 10254.

This research has also been supported by the Australian Government Research Training Program Scholarship managed by the UniSQ Graduate Research School.

DEDICATION

For my mother Bronwyn, my Aunty Jenny and Uncle Frank.

CONTENTS

LIST OF FIGURES

List of Abbreviations and Symbols

5G fifth generation.

ADC analog-digital converter.

AE auto-encoder.

AMC adaptive modulation and coding.

AWGN Additive White Gaussian Noise.

BER bit error rate.

BLER block error rate.

BP belief propagation.

BPSK binary phase shift keying.

CFO carrier frequency offset.

CNN convolutional neural network.

CSI channel state information.

DDPG deep deterministic policy gradient.

DDPM diffusion-denoising probabilistic model.

DL deep learning.

DQN Deep Q-Network.

DQPSK differential quadratic phase shift keying.

E2E end-to-end.

FFT fast Fourier transform.

GA genetic algorithm.

GAN generative adversarial network.

GEO geosynchronous equatorial orbit.

GPU graphics processing unit.

IDD iterative demapping and decoding.

IID independently and identically distributed.

IIoT industrial internet of things.

IoT internet of things.

IQ in-phase and quadrature.

ISI inter-symbol interference.

kNN k-nearest neighbours.

LDPC low-density parity-check.

LLR log-likelihood ratio.

LPD low probability of detect.

LPI low probability of intercept.

LSTM long-short term memory.

LTF long training field.

MAE mean absolute error.

MDN mixture density network.

MIMO multiple-input multiple-output.

MLD maximum likelihood decoding.

MMSE minimum mean squared error.

MSE mean squared error.

MUSIC MUltiple SIgnal Classification.

NN neural network.

OAL over-the-air learning.

OFDM orthogonal frequency division multiplexing.

PAPR peak-to-average power ratio.

PCA principle component analysis.

PG policy gradient.

PHY physical layer.

PSD power spectral density.

PSK phase-shift keying.

QAM quadrature amplitude modulation.

QPSK quadrature phase-shift keying.

RL reinforcement learning.

RNN recurrent neural network.

RTN radio transformer network.

SDR software defined radio.

SER symbol error rate.

SFE synchronisation feature estimator.

SGD stochastic gradient descent.

SINR signal-to-interference and noise ratios.

SISO single-input single-output.

SNR signal to noise ratio.

SOTA state of the art.

SPSA simultaneous perturbation stochastic approximation.

SSPA solid state high power amplifier.

STF short training field.

VAE variational autoencoder.

W-GAN Wasserstein generative adversarial network.

WSN wireless sensor network.

Chapter 1: Introduction

1.1 Background

Wireless communications are a substantial part of our digital communications platforms supporting broadband connectivity and networking for fixed, mobile and vehicular communications [1]. This form of connectivity is an important enabler for the formation of hybrid networks which extend connectivity to remote locations, leveraging hybrid terrestrial and non-terrestrial platforms, thereby enabling ubiquitous connectivity [1, 2]. Future communications technology will enable disruptive applications that continue to transform society, with broad applications and services including video, augmented reality, holographic communications, remote haptic control, transport and logistics and machine to machine communications [1, 2]. These applications are driven not only through the proliferation of mobile and smart devices but also due to the disruption of Industry 4.0 and the internet of things (IoT) [1, 2]. However, the capabilities required to support these applications place demands on wireless communications networks with increased requirements for data transfer, reliability, availability, privacy and security [2]. While fifth generation (5G) mobile communications already supports some of these capabilities, the changes required to support all of these new demands in future wireless communications systems, including at the physical layer (PHY), are still a subject of ongoing research [2].

The aim of a wireless communications system is to transmit a message over the air (the channel) to a receiver which can retrieve the original message from the received signal. The channel environment adds noise and distortions to the transmitted message and imperfections in electronic circuits causes additional perturbations to the signal such as timing, frequency and phase offsets. This means that the receiver must have the ability to correct imperfections to recover the original message.

Figure 1.1 illustrates a simple wireless communications system. In this setting the message M is represented as a sequence of K bits, (this allows for 2^K possible messages). To enable error correction in the receiver, the transmitter may optionally convert the K bits into a code of length N symbols. A code adds additional symbols to the transmitted message and as a result trades off the number of channel uses (and energy) required to send the message. This trade-off can be indicated by a code rate $R = K/N$ which is the ratio of message bits to the length of the code. The coded (or uncoded) message must be converted into a waveform to be transmitted over the channel. The modulation converts code symbols into a complex domain

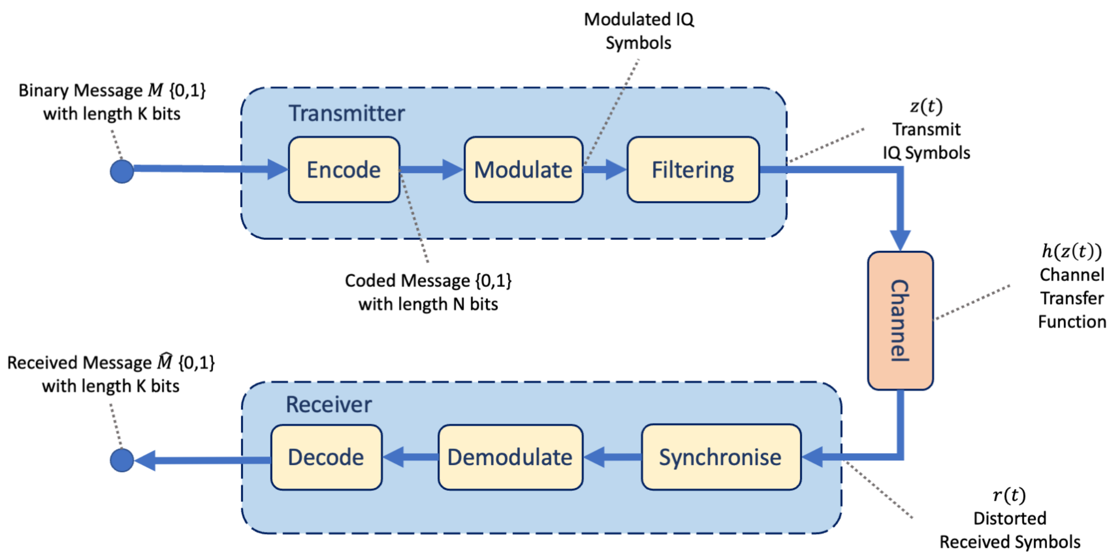


Figure 1.1: A simplified view of a wireless communications system. The transmitter takes in a K bit binary message M, codes, modulates the message and formats the resulting modulation as a waveform for transmission over a channel medium, producing transmitter symbols $z(t)$. These symbols are transmitted over a channel, represented as a channel transfer function $h(z(t))$, which produces noise and outputs symbols $r(t)$. The receiver performs synchronisation, corrects the channel distortions, and detects the signals carrying information. It demodulates each information frame and performs decoding to produce an estimate for the original message $\hat{M}$.

number $s \in \mathbb{C}$ comprising of in-phase and quadrature components (IQ). There are several forms of modulation to choose from which can use phase, amplitude and frequency or a combination thereof to represent the coordinates of the modulation symbols. For simplicity, the work in this thesis will refer to phase-shift keying (PSK) based modulation, which leverages the difference in phase for a constant amplitude in each symbol. Variations of PSK include binary phase shift keying (BPSK) and quadrature phase-shift keying (QPSK). The resulting waveform is often filtered to avoid inter-symbol interference (ISI) where the distortions due to the channel cause adjacent symbols to overlap with each other. A matched filter is applied to expand the width of each symbol and to enhance detection at the receiver. The resulting output of the transmitter is represented as $z(t)$ for each time-step t on the interval T. The channel environment is modelled as a transfer function $r(t) = h(z(t))$ which adds effects such as fading and represents the noise added by the electronic components in the transmitter and receiver. Other imperfections including hardware phase, frequency and timing offsets as well as Doppler frequency offsets are modelled as part of the channel transfer function. The received symbols $r(t)$ are synchronised at the receiver, synchronisation removes timing, phase and frequency offsets. The modulation used at the transmitter must be reversed to obtain the resulting code word, this process is known as demodulation. This code word is decoded and an estimate for the original K bit message $\hat{M}$ is produced by the receiver.

Traditional design approaches for wireless communications systems optimised each stage as individual signal processing blocks in isolation against an assumed and simplified mathematical model. The independent optimisation of each block does not guarantee optimal performance when combined into an entire system, and the new demands placed on wireless communications systems challenge this design process to achieve more optimal overall performance [3]. Researchers have recently begun to look to deep learning (DL) methods for their potential use as a design tool with the ability to jointly optimise the different stages in the wireless communications signal processing pipeline [3]. DL models are neural networks composed of many layers of non-linear activations which learn to extract meaningful representations from training data with respect to an objective loss function [4]. The success and increasing research adoption of DL in many fields such as computer vision and speech recognition has been attributed to several factors including advancements in hardware such as the graphics processing unit (GPU), the availability of labelled data sets and the development of open source frameworks supporting the development of DL models [4]. DL makes the assumption that the data is independently and identically distributed (IID), which imposes limits on the adaptability and generalisability of the technique [4]. Nevertheless, the composition of non-linear activations, combined with depth and the ability to train over enormous scale of data, due to supporting hardware and memory, have contributed to DL out-performing many previous state of the art (SOTA) methods in the many domains where it has been applied [4].

The adoption of DL in PHY wireless communications has been in application to the separate stages of the signal processing chain, as well as to the design of coding and modulation [5, 6]. Current research has demonstrated that DL methods can be incorporated alongside conventional signal processing blocks by unfolding the different iterations of the conventional algorithm [5, 6]. DL has also been applied as an end-to-end data-driven learning method which is capable of jointly optimising all stages of the signal processing pipeline [5, 6]. There is evidence that the DL method can outperform the conventional methods in channel environments which are highly non-linear and do not have an analytic mathematical model [5]. This points to the potential advantage of incorporating DL techniques to aid in the design of wireless communications systems. The trade-off between the use of an assumed model versus a data-driven learned model is a reflection of the bias-variance trade-off [5]. Suggesting that an assumed mathematical model biases the resulting system such that it performs well where the environment reflects the assumed model, but limits performance outside of such constraints [5]. Whereas a learned model achieves better generalisation when the training data set can incorporate a broad range of observations, reflecting the distributions of the real world [5]. The latter point however is also limited by the conventional supervised learning IID assumption.

This doctoral thesis is motivated by the application of artificial intelligence DL methods for the design of PHY wireless communications systems. In particular, the application of DL to synchronisation, over-the-air learning (OAL) and the learning of adaptive modulation and coding (AMC) schemes. Synchronisation of the wireless signal is challenging for DL and our work focuses on the estimation of the carrier frequency offset (CFO), which is introduced due to hardware timing imperfections as well as the Doppler effect in mobile networks. While the conventional signal processing techniques for CFO have been developed over many decades, DL is interesting because of it's potential application to a broad range of modulation schemes, including chaotic modulations used in physical layer security. OAL is significant for the end-to-end (E2E) learning of wireless communications systems over a physical channel environment. Learning in these environments has limited resources, therefore this research must consider the simplicity of the custom learning algorithm and the modelling approach. Adaptation over multiple channel conditions is also enabled by tuning in OAL approaches. However, this research also investigates how the DL architecture can be modified to enable learning different length codes for AMC, which can achieve spectral efficiency under changing channel conditions without the need for retraining.

1.2 Research Problems

The present thesis considers the notion that DL is capable of fusing high dimensional data and given an appropriate loss function and training algorithm, is able to learn mappings from non-linear data to supervised labels, or alternately, to approximate the distribution of the input data. However, there are several areas which are challenging for the application of DL as a design tool for wireless communications systems. Unlike other domains, there are not many canonical data sets available for research into DL for wireless communications systems [5, 6]. Instead, the data generation process is largely dependent on simulation, which in this work is primarily concerned with the generation of physically plausible signals, and channel distortions. The scope of the research in this doctoral thesis does not extend to the integration of these models in physical hardware, which is a consideration for future work. Instead, the significant publications included as a core contribution in this thesis describe the simulation of an appropriate data generation process for training DL architecture in each of the respective research problem domains. The research problem domains include Synchronisation, AMC schemes and OAL algorithms.

Synchronisation in the receiver is concerned with the removal of channel distortions from the received signal. A key aspect of this is the CFO, which may prevent demodulation of the received signal. Conventional estimation of the CFO relies on prior knowledge of the modulation and applies the fast Fourier transform (FFT) to detect the frequency peak of the received signal. The accuracy of the FFT is dependent on the length of the signal and the signal to noise ratio (SNR) [7]. In IoT communications, resource constrained devices rely on short communication blocks to limit the amount of energy consumed in communications and in turn impact the accuracy of the CFO estimate. An increasing concern in IoT communications is security. Chaotic modulations are advantageous for PHY security in energy constrained IoT devices since they avoid additional communication overhead of key sharing in application layer security [8]. These non-repeatable modulations are dependent on the sequential synchronisation of parameters used to generate the chaotic map. This process is dependent on removal of the CFO for the continuous non-repeating waveform [9, 10], which can be achieved by using fixed preambles of a non-chaotic sequence [11]. While a fixed preamble may be vulnerable to intercept by eavesdropping, it is desirable to maintain a short preamble length to maximise the channel use available for the information carrying signal. In energy constrained IoT devices it is desirable to estimate CFO for a variety of conventional and chaotic modulations, without an over-reliance on fixed preamble sequences. If this is achievable, it is possible then to maximise the energy allocated to transmit signals carrying information and minimise overhead required for pilot or preamble signals. This in turn increases data transfer rates in the emerging use of IoT and sensor networks in applications such as industrial manufacturing, personal area networks or health monitoring devices.

E2E training of wireless communications systems is proposed to enable a data driven joint optimisation of the transmitter and receiver with respect to the channel environment. This is contrary to the conventional block design of individual signal processing stages where each block is optimised individually against an assumed mathematical model of the channel environment [3, 5]. Conventional block design cannot guarantee globally optimal performance for the end-to-end system, and is dependent on the prior assumptions described by the adopted channel environment models [3, 5]. DL has played an important role in the E2E learning regime, however, E2E training is dependent on an assumed differentiable channel model [3]. The differentiable channel model is necessary to support the stochastic gradient descent (SGD) training approach where gradients are backpropagated from receiver to transmitter during the training process. However in an OAL algorithm, the transmitter and receiver are separate from each other and the gradients of the channel are no longer available [12, 13]. A key advantage in adopting DL methods is its ability to learn from complex data generation processes which may not have an analytic model. In wireless communications, the primary data generation process is the channel environment. For DL to be advantageous outside of well described analytic channel models, it is necessary to develop OAL methods which address the separation of transmitter and receiver and allow joint optimisation over true channel. OAL algorithms must consider the channel use required for feedback over the channel as well as the complexity of the modelling approach and training algorithm. Such algorithms have application to reliable communications systems, which can adapt to complex channel environments including adversarial environments which may experience intermittent accidental or deliberate interference. Additionally, the data-driven joint optimisation of transmitter and receiver allows the design of non-conventional codes, which are not only optimal for the channel environment, but may have a low probability of intercept (LPI) and thereby improve PHY security for wireless communications.

While OAL algorithms can enable both the DL based transmitter and receiver to be tuned under changing channel environments, there is a cost in terms of the energy, time and channel use required to tune these models on device. Meanwhile it is necessary to support ongoing communications, potentially over a redundant link, thereby requiring additional hardware. Conventional approaches which perform AMC are able to optimise communications over varying channel conditions [14, 15]. AMC maps a set of modulation and coding schemes to minimise the expected error-rate under changing channel conditions and transmit power [16]. E2E wireless communications systems trained with DL have the limitation that the DL architecture is capable of learning only one code rate K/N. To achieve multiple code rates, it is necessary to train multiple configurations of the DL transmitter and receiver. This is a different problem to accommodating different length inputs such as via a convolutional neural network (CNN) module. Instead, the transmitter must be able to learn to output a different code of length N given both the message M and some context information. Learning AMC requires modification of the DL architecture and the E2E training algorithm to support multiple code rates. The ability to do so will support reliable wireless communications and improved data rates when the channel environment changes. A data-driven methodology such as DL would enable further integration opportunities such as in learning the optimal control of selected code rate to the continuous variation of the channel environment. A key advantage of learning an AMC with DL is that it would mitigate the need for retraining the transmitter and receiver when the channel environment changes, thereby achieving adaptation in wireless communications systems without the cost of model tuning.

1.2.1 Research Aims and Objectives

This doctoral thesis is motivated by the challenges described for each of the research problem domains of Synchronisation, OAL algorithms and AMC schemes for the E2E development and design of wireless communications systems. The objectives for this research are presented in four publications comprising the body of this PhD thesis by Publications. Each publication addresses one of the focus areas and make the following contributions to the growing interdisciplinary field of AI-enabled communications.

1. **Objective 1. Synchronisation.**

    Enhance the security of IoT burst communications by developing a stacked sequence-to-sequence DL estimator for the CFO estimation problem and support blind estimation of CFO for multiple conventional and secure chaotic waveforms. Blind CFO estimation is a step towards enabling E2E communications systems which operate in a physical channel environment.

This research output is presented in Chapter 4, and has been published in IEEE Access (vol. 10 (2022), pp 119814-119825, ISSN 2169-3536).

2. **Objective 2 Adaptation.**

   Enable adaptive communications in E2E training of wireless communications systems. This is achieved by modifying the DL architecture to support multiple code rates, and to frame the challenge of learning multiple code rates as a multi-task learning problem.

   This research output is presented in Chapter 5 and has been published in Applied Soft Computing (vol 159 (2024), p 111672, ISSN 1568-4946).

3. **Objective 3. Over-the-air learning (OAL) with feedback.**

   Develop novel approaches to training wireless transmitter and receiver OAL and improve on existing methods by simplifying the training procedure. This research aims to demonstrate that it is possible to simplify training of the transmitter by learning implicit information about the channel through the errors made at the remote receiver.

   The research output is presented in Chapter 6 and has been published in the journal MDPI Sensors (vol. 23 (2023), no. 24, p 2848, ISSN: 1424-8220).

4. **Objective 4. Over-the-air learning (OAL) without feedback.**

   Enhance privacy of data-driven wireless communications systems by developing OAL algorithms supporting training without use of a feedback channel as well as simplify channel modelling approaches for use in E2E training. The work enhances the security of OAL by training a channel model on uniform noise transmitted through the true channel environment. The aim is to develop an efficient channel model which can accurately model a variety of channel environment distributions. The resulting channel model supports training transmitter and receiver on the remote side of the network and avoids information carrying transmissions during the learning procedure. Thereby reducing the opportunity for adversarial intervention during learning.

   This research is presented in Chapter 7 and has been published in the journal MDPI Sensors (vol. 24 (2024), no. 10, p 2993, ISSN: 1424-8220).

1.2.2 Significance of Research

Objective 1. Synchronisation

Research contributions for the Synchronisation research focus are developed in Chapter 4 and are highlighted below:

- Develop a stacked sequence-to-sequence DL model which gradually refines the estimate for the CFO for conventional BPSK and QPSK modulations as well as for the Circular, Quadratic and Zadoff-Chu chaotic map modulations.
- Evaluate the use of discretisation of the target frequency for the refined estimation method of the model against the use of continuous regression of the target frequency.
- Compare the accuracy of the proposed model CFO estimate on the received signal with and without data augmentation, and demonstrate that the addition of data augmentation enhances the accuracy of the model.
- Compare the proposed model with several conventional FFT based CFO estimation methods for conventional modulations as well as correlation methods for chaotic modulations, with comparisons made both for mean absolute error (MAE) and execution timing.

Objective 2. Adaptation

Adaptation of E2E for wireless communications systems is introduced in Chapter 5 which develops an E2E training method AMC and makes the following contributions:

- Proposes an end-to-end machine learning architecture for generating coded modulation schemes with different data rates.
- Custom training/multi-task learning produces coded modulation schemes with competitive error-rate performance.
- Proposed approach outperforms several traditional coding techniques for short codes.
- Proposed approach is versatile in adapting to Rayleigh fading channel condition without model retraining.
- The proposed end-to-end machine learning architectures have practical benefits for developing resilient wireless communication systems.

Objective 3. Over-the-air learning (OAL) with Feedback

Research contributions for the development of OAL algorithms are developed in two articles. Those developed in Chapter 6 make contributions in the following areas:

- To propose a novel over-the-air training method and develop machine learning enabled coding and modulation schemes for the transmitter and the receiver without an assumed channel model.
- To develop a Disjoint Learning method that uses a transmitter-side (local) channel/receiver to imitate the learning process of the remote receiver and enable supervised learning of the transmitter through backpropagation.
- To demonstrate that the performance of the proposed Disjoint Learning method is equivalent or better than the fully connected architecture.
- To show that the proposed method achieves significant performance improvements against the Receiver Tuning training method.

Objective 4. Over-the-air learning (OAL) without Feedback

The research presented in Chapter 7 extends the work on OAL algorithms with the following research contributions:

- The research article proposes an iterative OAL algorithm for the development of transmitter, receiver and channel model which does not require continuous feedback between transmitter and receiver. Thereby reducing the vulnerability to eavesdropping and adversarial attacks during the learning procedure.
- The research presents application of the mixture density network (MDN) for the approximation of the channel transfer function. And demonstrates the approximation for several simulated channels including the Additive White Gaussian Noise (AWGN), Rician fading, Rayleigh fading and power amplifier AWGN channels.
- The article demonstrates that intermittent measurement of block error rate (BLER) for transmitter and receiver models, using the generative channel model instead of the true channel environment, is suitable for use as the training stopping criteria and for monitoring of learning process.
- Finally, the research article shows that the performance for the resulting transmitter and receiver models are equivalent to or better than the end-to-end model which is trained with an assumed channel model. This is shown for AWGN, Rician fading, Rayleigh fading and non-linear power amplifier with AWGN simulated channels.

Both of the novel methods developed for OAL learning are able to be applied to adaptation of DL based wireless communications systems. However as noted, tuning of the transmitter and receiver has a cost in terms of both time and energy on resource constrained devices, whereas the approach developed for AMC serves to provide adaptive communications without the requirement for tuning.

1.3 Thesis Layout

This thesis is structured in seven chapters, and includes four published articles on the use of DL for data-driven design of Synchronisation, Adaptive Modulation Coded Schemes and OAL algorithms.

Chapter 1 provides an introduction to the topic and briefly introduces the reader to the significance of wireless communications, as well as describing a simplistic design for a wireless communications system. It introduces the use of DL for design of wireless communications systems, and describes the research domains which are the focus of this thesis.

Chapter 2 presents a survey of literature relating to the application of DL for the design of wireless communications systems. It enumerates the applications of DL for the three thesis objectives of Synchronisation, Adaptation and OAL algorithms for wireless communications systems and identifies gaps in the research domain along with the motivations for the research focus areas presented in this thesis.

Chapter 3 describes the data generation approaches and methods of simulating channel environments which are applied in the experimental approaches for the published research. This is a summary of the relevant data generation and simulation methodologies that are contained in greater detail in each of the featured publications.

Chapter 4 (*research publication 1*) describes the development of a novel DL architecture for the blind CFO estimation problem for two conventional modulations and three chaotic modulations. This chapter reports the findings of Objective 1.

Chapter 5 (*research publication 2*) presents an original research article demonstrating a novel DL architecture and training algorithm for the E2E learning of AMC schemes. This chapter reports the findings of Objective 2.

Chapter 6 (*research publication 3*) proposes an algorithm for OAL which is simpler than current methods described in the literature and demonstrates that it is possible to learn implicit information about the channel environment without modelling it explicitly. This chapter reports the findings of Objective 3.

Chapter 7 (*research publication 4*) further develops an algorithm for OAL which does not require feedback over the channel. It demonstrates that it is possible to learn an approximation for the true channel environment without an information carrying signal. And applies the developed channel model to the E2E training of transmitter and receiver without exposing a feedback channel to potential adversarial attacks. This chapter reports the findings of Objective 4.

Chapter 8 summarises the research findings of objectives 1-4, reported in four different publications, and discusses the limitations of the research work and further proposes future directions for this area of research.

Figure 1.2 presents a schematic overview for the body of this doctoral thesis excluding the Chapters for Introduction, Literature Review and Conclusion.

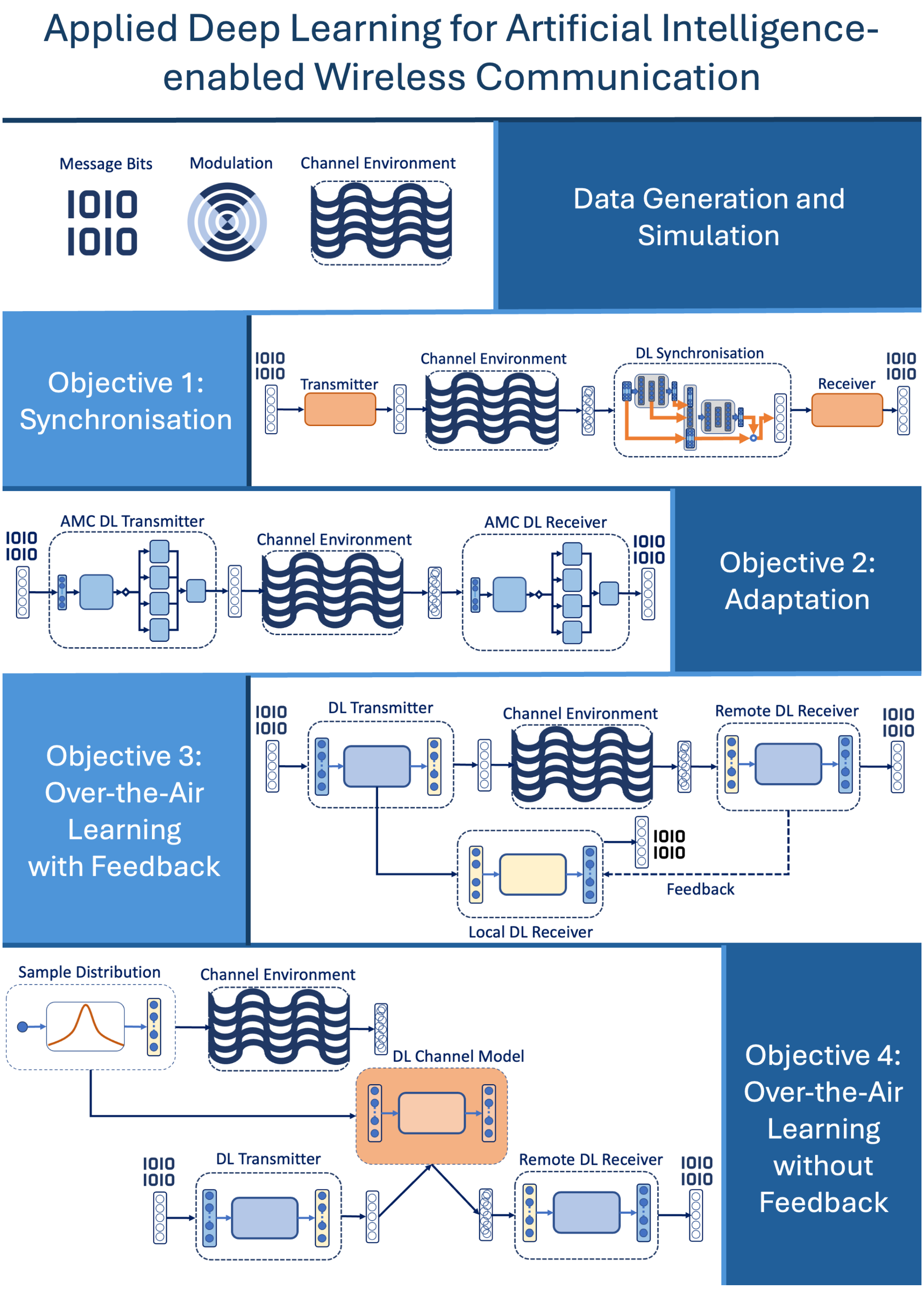


Figure 1.2: An overview for the main topics comprising the body of this doctoral thesis. These include: Data Generation and Simulation; Objective 1: Synchronisation; Objective 2: Adaptation; Objective 3: Over-the-Air Learning with Feedback; and Objective 4: Over-the-Air Learning without Feedback.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Research interest in the application of neural networks (NNs) to the modelling of digital communications systems has developed over the past thirty years [17]. The primary argument given in the literature for the use of DL in wireless communications is that wireless communications systems are conventionally designed via a block design approach. Each block or component responsible for a signal processing task is optimised for an assumed mathematical channel environment, and is often designed in isolation from each other [3, 17, 18]. Due to this localised optimisation, global optimisation of the system cannot be guaranteed [3, 18]. DL offers the potential for a data-driven approach that is able to optimise all components from observations of the channel environment, and optionally, without assumed knowledge of the channel [12, 19]. Hence NN and DL exhibit the ability to learn non-linear characteristics of the physical channel environments where it has been challenging or intractable for analytic methods [17]. In end-to-end optimisation this learnt information is leveraged by the transmitter to learn a code representation that is able to improve the performance of the receiver [3, 18]. While in the receiver, DL is able to learn transformations capable of denoising, correcting channel imperfections and detecting transmitted symbols to improve performance in recovery of the original message [3, 18]. In wireless communications DL is leveraged both as a task specific subcomponent to estimate parameters in the signal processing chain, such as in synchronisation [20, 21], or otherwise as a design tool where many or all components are optimised jointly [3, 18]. In both cases, the channel is either modelled as a stochastic process during simulation or is a physical process that is observed directly. Regardless of the scenario, there are key challenges to complexity, generalisation and adaptation given the application constraints, where the literature proposes solutions comprising approaches to data preparation, model architecture and training algorithms.

2.2 Parameter Estimation for Synchronisation

Synchronisation is a task where DL has been applied to model one or more parameters or perform transformation of the received signal in an end-to-end process. In a wireless communication system, there are several imperfections which arise out of the physical channel environment as well as the electronic components. Mismatch in oscillators between the transmitter and receiver result in carrier frequency offsets and phase offsets between the transmitted and received signal. Propagation delay results in timing mismatches between the transmitted signal and the received signal. Each of these offsets require correction at the receiver prior to demodulation. These imperfections manifest as perturbations of the signal waveform, which in its simplest expression can be described as a sinusoidal waveform. Optimal conventional methods, assuming an AWGN channel, have been developed over a long history for phase, frequency and timing parameter estimation and correction [21–23]. Nevertheless the application of DL to the synchronisation problem has been demonstrated to exhibit advantages to conventional methods given constraints of sample length or in low SNR [21–23].

The task for estimation of the frequency is reduced to its simplest form in [24], where a fully connected network is used to estimate the frequency of a sinusoidal wave with additive Gaussian noise (at a SNR of 25 dB). A NN was trained to estimate the normalised frequency offset between 1 kHz and 10 kHz through regression for an input sequence of 2000 samples. This approach demonstrated that it was possible to estimate the frequency component with good accuracy having generated enough training samples. It is a highly simplified example, without variation of SNR, it is not compared against conventional methods for frequency estimation and is not applied to a modulated waveform as would be used in a wireless communications system.

A smoothed estimate of the frequency pseudo-spectrum for a complex waveform is produced by training a mixture of fully connected and convolutional NN layers in [25]. The proposed method PSnet estimates a smoothed frequency pseudo-spectrum from a fixed number of samples (5000 samples for the input layer) where the waveform may contain multiple frequencies separated by a small increment [25]. The data generating process in this work is closer to a communications use case, although not explicitly designed for communications. The training waveforms are generated for variation in amplitude, and frequency and the additive Gaussian noise is sampled at several SNR between 1 and 100 dB [25]. The authors speculate that the first layer of the NN is responsible for translation from a time domain into a frequency domain, and acknowledge the learning constraints in the NN due to the IID assumption for the generated training data set [25], noting that the distribution from which the training data is drawn influences the overall accuracy of the network. This has ramifications for the communications setting especially where the number of bits in a message frame would cause sampling from a broad number of messages and channel conditions to become impractical. The proposed PSnet did demonstrate better accuracy than the conventional periodogram and the eigenvalue based MUltiple SIgnal Classification (MUSIC) estimator in low SNR [25]. However, the study did not focus on the communications use case, such as the effect of a communications modulation and filter on the performance of the model.

Several DL techniques are trained to estimate CFO and compared against conventional methods for a wireless communications application in [22]. In particular, the quantisation that is applied to simulate a 1-bit analog-digital converter (ADC) causes a loss of information but is advantageous for energy constrained systems such as IoT devices [22]. The article experimentally trains NN, CNN, long-short term memory (LSTM) and residual CNN for short block lengths of 8, 16 and 32 bits in SNR between 0 to 10 dB. [22]. It compares performance on both quantised and non-quantised waveforms and evaluates performance against the periodogram, MUSIC and Welch's method for estimation of CFO [22]. The results reflect good performance for CNN and LSTM based networks trained in low SNR and comparable performance to conventional methods in high SNR. In addition the accuracy of the DL models is shown to improve as the block length increases [22]. In both [25] and [22] complex in-phase and quadrature (IQ) values are treated as two separate features for the in-phase and quadrature components, since the DL methods are not designed to process complex valued data. This article does provide some evidence for the ability of DL to estimate CFO for short blocks. Although the approach is quite exploratory, the data generated is close to a PSK modulation but is a random sequence, and like the other articles does not investigate multiple kinds of communications modulations. Structure within communication preambles or modulation may influence the features learnt by the DL estimator.

Data-driven preamble detection is performed by a 1d-CNN which learns to produce a soft ranking for indices of a complex correlation in [23]. In this way the CNN is applied to determine the alignment between a reference preamble and arriving packets and rather than relying on a single maximum value for the peak of the correlation, it can make use of the distribution of the correlation window to more accurately estimate the beginning of the preamble [23]. The method is shown to exhibit a lower number of false detections than the conventional correlation based estimator in both AWGN and a flat fading channel [23]. By using the complex correlation for input to the DL model, the article demonstrates how a data-driven method can be integrated with a conventional method to achieve improvements in accuracy. The preambles leveraged in the experiments consisted of random BPSK encoded data, however other kinds of preamble such as those based on Barker codes [26], chirp [27] or Zadoff-Chu sequences [28] exhibit excellent correlation properties which improve the performance of the conventional method. The article also did not examine the impact of modulation on the parameter estimation problem.

The 1d-CNN has also been demonstrated as an effective approach to learning filters for preamble detection in [29]. Preambles generated from pseudo-random noise were evaluated at lengths of 8, 16 and 32 bits in AWGN, Rayleigh and Rician fading channels and were combined with additional perturbations for phase offsets and CFO [29]. False Detection rates were compared with a correlation method featuring adjustments for the additional perturbations, and the CNN filter demonstrated favourable performance in shorter length preambles with the conventional correlation method performing similarly on the 32 bit preamble [29]. The CNN is trained only on the AWGN channel and evaluated on both fading channels without retraining, and is shown to be more accurate for the shorter preambles than conventional methods in both of the fading channels [29]. Like the experimentation in [23], this article does not examine preambles with optimal correlation properties. While it does demonstrate good performance for the data-driven method, the use of optimal preambles such as those with low auto-correlation ambiguity would enable optimal performance from conventional correlation detection methods [30]. Comparison under such conditions would determine whether data-driven methods have the ability to automatically learn and benefit from properties such as optimal auto-correlation of the input signal.

A realistic application of DL to wireless communications features two variations of a recurrent neural network (RNN) to estimate CFO and a 1d-CNN for frame detection of an IEEE 802.11 orthogonal frequency division multiplexing (OFDM) communications system in [21]. The packet contains short and long training fields (STF and LTF) which are used for coarse and fine parameter estimation respectively [21]. Conventional detection is performed with complex correlation for a sliding window against a known preamble sequence, preambles are predefined such that they have good correlation properties for the beginning of the sequence. While the conventional CFO estimator calculates the phase of complex correlation for successive symbols and produces a coarse frequency estimate from the STF and a fine frequency estimate from the LTF. A final estimate is produced by adding both estimates together. The DL models are trained by simulating the channel environment with a range of SNR and perturbations, as well as by capturing observations from a physical lab environment using software defined radio (SDR) [21]. The 1d-CNN is more accurate than the conventional detector on short sequences up to 320 samples under simulation, and performs similarly in the real environment [21]. The conventional CFO estimation is more accurate than the DL CFO estimator in high SNR above 8 dB, with the DL being more accurate below that [21]. The conventional CFO estimation method benefited from a two stage estimation process, making coarse and fine grained estimates. The DL method was not designed to refine the estimation however it is possible to design an architecture that may perform some kind of refinement during the forward pass such as through upsampling and interpolation. Both tasks also benefited from a fixed preamble, due to the frame design however, it is also possible that preambles may be required to have dynamic properties which would require training the DL models either with broader examples, or to learn have some generative capacity to address the generalisation challenge. In addition, the channel environments presented in the article are relatively simple, while the real lab scenario does present a multi-path fading environment, it is uncertain how well the proposed method would perform under a number of different channel conditions.

The applications of DL for synchronisation parameter estimation presented so far demonstrates a framing of the problem where the model is relatively isolated from the entire signal processing chain, and the parameter estimates may be used in a more conventional manner when correcting the received signal. However, this does demonstrate the capacity for DL methods to match or outperform conventional methods which must estimate properties of the perturbed signal for a given channel environment. Much like the block design approach in conventional systems, DL can be composed into a larger system. The key difference, is that once organised as part of a larger architecture, all components (or layers) in the model can be optimised with respect to the channel environment, constraints and a loss function. This approach gives rise to the pursuit of End-to-end learning for wireless communications in the research literature, which takes advantage of the auto-encoder (AE) architecture in place of transmitter and receiver. The AE is a form of unsupervised learning that is able to learn low-dimensional features which are not only suited to reconstruction, but also serve as a mapping between input and a coordinate system that is similar to non-linear principle component analysis (PCA) decomposition [31]. In the case of wireless communications, the features produced by the encoder correspond to the learnt code space.

2.3 End-to-End Learning

The canonical use of the AE for E2E training of a wireless communications system is presented in [3]. A simplified communications system is developed to learn short codes (the Hamming$(7,4)$ code and uncoded 8-bit BPSK) subject to an energy constraint and over an AWGN channel [3]. The AE network serves as the reference architecture, consisting of a transmitter including an energy constraint (the encoder), a channel model which is a non-trainable distortion applied to the transmitter output, and the receiver (the decoder). The system is trained to map a one-hot encoded representation of the message into a code representation of size N where N complex IQ values are represented as two real vectors. A one-hot encoded representation for K bits is a vector of length 2^K where each position represents one of the unique 2^K bit messages. Hence each K bit message is encoded as a unique symbol (symbol-wise representation) rather than a sequence of bits (bit-wise representation). The receiver learns to predict the probability for each of the 2^K messages. The role of the receiver is to perform classification of the received signal that has been distorted by the channel layer. Since the AE is optimised E2E, the transmitter learns an optimal code for the given channel distortion enabling the receiver to maximise classification performance. The AE is demonstrated to achieve equivalent BLER to the maximum likelihood decoding (MLD) for the Hamming$(7,4)$ code and produce gains over uncoded 8-bit BPSK in the AWGN channel [3]. The joint learning capability is also demonstrated by training two AE over an interference channel so that the output of opposing transmitters are added together, and each receiver must classify the original message from its respective transmitter given the channel noise and superimposed codes. The resulting system automatically learns an orthogonal code for each respective transmitter receiver pair and therefore minimises the impact of the interference signal [3]. A key limitation of the approach is that the channel transfer function represented by the non-trainable channel layer must support differentiation to permit the transfer of gradients through the network during training by back-propagation [3]. The symbol-wise representation is also limited by an issue of scale, since the increase in symbols is impractical for longer K-bit messages [3]. The training procedure is negatively impacted by the ability to sample messages as well as the dimension required for the input and output at transmitter and receiver.

The limitation of the symbol-wise approach is addressed by training an AE to learn both a coded constellation at the transmitter and a soft estimate for each bit at the receiver in [32]. Both transmitter and receiver also take as input the estimated SNR for the channel and the resulting AE is shown to outperform the bit error rate (BER) of both PSK and quadrature amplitude modulation (QAM) modulations in AWGN [32]. The availability of the SNR to both the transmitter and receiver enables the AE to learn an adaptive constellation with respect to the channel environment [32]. The soft values produced by the receiver are compatible with use in an iterative demapping and decoding (IDD) algorithm, such as belief propagation (BP) since they are estimates of the log-likelihood ratio (LLR) for each respective bit, this enables the use of the AE with an outer low-density parity-check (LDPC) code [32]. While the AE itself is limited to short messages (up to 8 bits) the outer LDPC code is leveraged to process message length of 1944 bits. An optimisation algorithm which uses the inner AE code is developed to select the optimal outer LDPC code and the resulting system is compared with a baseline LDPC code of the same length [32]. When used as an inner code the AE enables the outer LDPC code to achieve lower BER than the baseline [32]. The OAL version of the training procedure is also presented where the system is trained on SDR leveraging a method devised in [33]. In the OAL context the system demonstrates gains due to the ability of the AE to optimise for the channel environment and the optimisation process enables the IDD procedure to access more information in the received LLR sequence [32]. While the article presents an almost complete system, it's ability to extend the length of the message is largely due to the outer LDPC code, while the AE transmitter produces 2^N symbols which are constrained to shorter codes. The E2E experiments assume the AWGN channel environment, and portability between the assumed AWGN environment and the OAL channel is not examined, however results are indicative of the benefit of OAL.

Phase offsets and timing offsets are addressed in [34] by adding a radio transformer network (RTN) as a method of regularisation for the canonical AE. The RTN is derived from [35] and consists of both a localisation module for parameter estimation and an affine transformation module which transforms the received signal with the learnt parameters [34]. The AE is developed to transmit and receive a 128 bit message over a simulated AWGN layer and compared with the BER of both QPSK and 16 bit QAM modulations [34]. Several architectures and loss functions are evaluated and linear dense networks are shown to produce lower BLER than the baseline modulations [34]. The size of the input space is extremely large, 2^{128} and it is not possible to sample all points for training, especially when considering additive noise due to the channel layers. The results suggest the ability of the network to generalise given the improvement in BER over QPSK [3]. Although the experiments are limited to an AWGN channel environment. While the article does demonstrate the impact of phase and timing offsets on the performance of the network, the benefit of the synchronisation provided by the RTN is not fully demonstrated in comparison to the baseline modulations.

Performance evaluation of a proposed synchronisation feature estimator (SFE) for phase offsets, attenuation and timing offsets is made in comparison to theoretical BPSK for the AWGN channel [36]. The SFE provides a correlation like operation by combining CNN, pooling and dense layers as part of the receiver and the learnt features are concatenated with the channel output to perform classification at the receiver [36]. The proposed approach makes use of a symbol-wise encoding for the message and interleaves pilot symbols with information symbols in order to provide reference information for the SFE to compensate for phase offsets, attenuation and timing offsets [36]. The model is trained E2E with an AWGN channel layer and the additional perturbations, and several code-rates are evaluated in simulation as well as OAL post training. Use of the SFE is shown to be advantageous in comparison to the AE trained without it [36]. After training the transmitter and receiver models are deployed on SDR and demonstrated an error free data rate of 0.5 Mbps [36]. The addition of interleaved pilot symbols increases the channel usage reducing the code rate by $1/2$, which is an added complexity.

Conventionally pilot symbols are used to estimate channel parameters which can be used to assist with the recovery of the original message, which is distorted through fading and delay effects [18]. Instead of explicitly including pilot symbols in the transmitted signal, an E2E architecture is developed to implicitly learn robust constellations at the transmitter as well as extract channel features at the receiver in [18]. The approach leverages CNN modules to encode and decode bit streams of varying length and estimate channel features for a single selective flat fading channel with a 128 bit block size, as well as multiple-input multiple-output (MIMO) channels with a 256 block size, and is compared with QPSK modulation combined with OFDM [18]. The model is demonstrated to outperform minimum mean squared error (MMSE) decoding in the signal channel and performs close to MMSE decoding in the MIMO channel without perfect channel state information (CSI) [18]. Application to joint source coding is also demonstrated, where instead of coding image data to a bit representation, images are fed directly into the E2E model which learns to recover them at the receiver with higher accuracy than conventional image compression techniques [18]. The learnt transmitter and channel extraction features learnt by the E2E model are dependent on the assumed channel model and distribution of channel parameters, which is a limitation inherent to the E2E training approach since it is dependent on the assumptions made for simulation of the channel.

E2E training of wireless communications systems has been demonstrated in the literature to not only outperform simple block codes [3, 34], but also to incorporate concatenated coding such as in [32] and incorporate mechanisms in the model architecture to enable learnt synchronisation [18, 36]. However, the key limitation of the E2E approach is the requirement of a differentiable assumed channel model [3]. To this end, OAL seeks to develop methods which can train a transmitter and receiver without necessarily assuming a channel. Such methods extend E2E and the AE architecture by devising extensions which enable transfer learning in the receiver to adapt to a true channel environment (Receiver Tuning); learning a DL channel model through observation (Channel Approximation); using perturbation methods to approximate the gradient during backpropagation (Gradient approximation); or relying on coordination between multiple agents to enable learning between several instances of transmitters and receivers (Collaborative Agent Learning).

2.4 Over-the-Air Learning

Once a transmitter and receiver have been trained E2E over an assumed differentiable channel layer, the most pragmatic approach to OAL is to tune the receiver model in the true channel environment. Receiver Tuning is inspired by transfer learning in computer vision, where pre-trained CNN models may be adapted to new tasks by training only the upper layers of the model [37]. An AE is trained E2E on a simulated AWGN channel in [37], which includes pulse shaping, phase offsets, CFO and timing offsets. After training, the transmitter and receiver models are deployed in a hardware SDR environment and the receiver is fine-tuned for operation in the new environment [37]. Extensions to the E2E model are added to support pulse shaping as well as the additional channel perturbations. ISI is addressed by interleaving short information blocks with a small amount of padding, phase offsets and CFO are addressed by a phase estimator module inspired by the RTN in [34], and additional offset estimator and feature extractor blocks estimate sample offsets to perform frame synchronisation [37]. The outputs of each estimator are concatenated together as inputs into the receiver decoder. The system is trained in a symbol-wise manner to transmit an 8 bit message and the transmitter develops constellations which automatically include symbols for phase offset synchronisation [37]. The performance of the system is shown to be approximately 1 dB worse than differential quadratic phase shift keying (DQPSK) after fine-tuning on both the simulated and over-the-air channel environments [37]. Even though no gain is achieved against the baseline, the approach demonstrates the feasibility of designing a complete wireless communications system offline E2E and performing Receiver Tuning on the true channel post deployment. The primary disadvantage of Receiver Tuning is that the transmitter is unable to adapt the constellation that was learned via offline simulation to the true channel environment.

The main difficulty preventing OAL for the transmitter is due to the way DL models are trained with backpropagation. To achieve backpropagation, gradients must be computed for the change in weights with respect to the error in the model estimate after each forward pass and are used to update weights during the backward pass of each step. The true channel environment separates the transmitter and receiver. Therefore, a method of estimating gradients is necessary in order to train the transmitter in this context. Gradient Approximation achieves this outcome by leveraging methods based on finite difference approximation and reinforcement learning (RL) to estimate the gradient at the transmitter, given the receiver loss.

A variation of finite difference approximation is used to estimate the gradient at the transmitter in [38]. The receiver is trained with backpropagation, while simultaneous perturbation stochastic approximation (SPSA) is applied to train the transmitter [38]. The article simulates an AWGN and Rayleigh fading channel to compare against theoretical QPSK and the canonical E2E method devised in [3]. However, the training phases for the proposed method are separate for transmitter and receiver [38]. Equivalent performance is demonstrated in both channels between the proposed method and the canonical AE and is slightly worse than theoretical QPSK in the Rayleigh fading channel. However, SPSA requires feedback from the receiver loss for multiple perturbations of the transmitter outputs to calculate the gradient and results in a longer training duration due to multiple forward passes through the receiver. The method is also shown not to scale well when the number of parameters in the transmitter are increased [33].

A policy gradient (PG) method is applied to train the transmitter in [12, 33] where a two-step algorithm trains the receiver and samples perturbed outputs from the transmitter to estimate the gradient from feedback of the receiver loss. The primary difference between the SPSA method and the PG method relate to the way the gradients are estimated, the distributions for transmitter perturbations vary between the two methods and the latter method requires only one additional forward pass to estimate the gradient [12]. The method is evaluated on both AWGN and a Rayleigh fading channel against the supervised AE approach devised in [3]. In the Rayleigh fading channel, alterations to the design of the receiver network are required to compensate for the fading effects [12]. In [33], the proposed approach was compared with QPSK in each channel and its application in joint source channel coding was demonstrated for coding of images in an AWGN channel. While the proposed method takes longer to train, it exhibits equivalent performance to the supervised AE [12]. The convergence of PG methods are known to be impacted negatively where the reward signal exhibits high variance [39], with subsequent methods proposing sampling from a memory buffer to minimise variation at the cost of additional complexity [40]. The PG method is also applied in a more complete example incorporating IDD in [32] and subsequent work attempted to address the issue with variation due to feedback by incorporating deep deterministic policy gradient (DDPG) based methods in [41]. However, the primary issue with DDPG is the additional complexity that is introduced to the training procedure due to retaining previous transmission outputs and their values for use in sampling during training. The requirement for a reliable feedback channel for receiver loss is also an overhead that increases channel usage.

An alternate approach in addressing unknown channel gradients is Channel Approximation which models the channel by training a separate DL model and leverages the developed channel model in training the transmitter and receiver in an E2E manner. In this case the developed channel model provides a differentiable proxy of the true channel environment. The aim of this task is to develop a generative channel model which can accurately approximate the distribution of the true channel perturbations. In this framework several variations of generative model have been adopted including the variational autoencoder (VAE), generative adversarial network (GAN) and diffusion-based models.

Channel approximation with the GAN is demonstrated in [42] and the framework is later extended to use a variational generator in [43]. In [42] an iterative approach trains the channel model and uses it to train the E2E model over SDR, but does not leverage a discriminator model to train the channel generator, instead training the generator directly against the true channel outputs using mean squared error (MSE) loss. A Gaussian noise layer is required at the input of the generator in order to learn a constellation similar to 16-QAM [42], however the performance of the system is not compared with any conventional method. The channel generator in [43] is trained with a discriminator and rather than adding noise at its input, the generator is instead modelled after the VAE sampling from Gaussian noise internally. In this manner, the generator is demonstrated to approximate a range of simulated channel environments. The internal sampling layer is shown to be necessary to provide variation in the channel generator model as opposed to attempting to learn the distribution directly without a discriminator [43]. However, without evaluation such as through comparison of performance against a conventional modulation, the results are only illustrative, suggesting that a GAN appears to learn a good approximation of the true distribution.

A more thorough investigation is given in [13] where conditional GANs are trained to approximate AWGN and Rayleigh fading channels. The generator receives Gaussian noise and the addition of the transmitter output as conditional inputs, and is trained against a discriminator which also receives the transmitter output to learn how to imitate the true channel [13]. The training algorithm is iterative where the channel is updated while the transmitter and receiver are frozen in the first iteration, and transmitter and receiver are updated while the channel is frozen in the second iteration [13]. Once trained the transmitter and receiver are evaluated over the simulated channel functions and shown to perform similarly to the Hamming$(7,4)$ code in AWGN and 16 QAM in the Rayleigh fading channel with the addition of pilot symbols for conditioning [13]. A later article [19] extends the approach to a longer bit length of 128 bits in AWGN and 64 bits in Rayleigh and frequency selective fading through the use of CNN in the transmitter and receiver. The approach demonstrates the ability to train a GAN on a simulated channel and use it to train transmitter and receiver E2E in a simulated environment, however it is unclear how the proposed method would work in a physical environment. In a physical environment a feedback channel may be required to relay channel outputs to the GAN to enable training in an iterative manner on the transmitter side.

A feedback channel is not always required if the model used for Channel Approximation can be trained with outputs from the true channel environment on the receiver side and later used in E2E training. A conditional Wasserstein generative adversarial network (W-GAN) is trained to approximate the channel and develop a transmitter and receiver with a custom training algorithm (one-shot training) in [44]. The one-shot training is achieved on the receiver side and develops the W-GAN channel model using transmitter output sent by a pre-trained AE based on the IDD method described in [32]. Once the channel model has been trained, the AE is trained by using the W-GAN as the channel proxy [44]. The approach is trained OAL using SDR in a similar manner as [32], and demonstrates equivalent performance to the RL method. While the proposed approach does not require feedback, the use of the pre-trained AE to train the W-GAN does require some prior knowledge of the channel environment, it would be desirable for a Channel Approximation method to be developed without an assumed modulation. An additional consideration for GAN methods is that the adversarial training method requires an addition of the discriminator model, which is discarded after the generator has been trained, which introduces a level of complexity during the training routine. Another disadvantage of GAN channel generators is that the GAN is known to suffer from mode collapse, which is a general problem where a generator may learn to fool the discriminator using a restricted set of outputs and not approximate the entire distribution [45].

While the W-GAN approach introduces constraints and customises the loss to help improve generalisation and reduce mode collapse, other generative methods have been explored to further improve generalisation over the channel distribution. In [46] a diffusion-denoising probabilistic model (DDPM) is trained to approximate the distribution of several channels. A diffusion model learns a forward noising and reverse denoising process and is able to generate samples from the denoising process in an iterative manner [46]. However the iterative procedure is slow to produce samples during denoising, as the model must be invoked multiple times. To address this the authors propose a method which relies on a subset of the sampling iterations [46]. Two approaches to training the wireless system are proposed, the first pre-trains the generative model on random symbols transmitted through the true channel environment and uses the resulting generative model in the E2E learning for transmitter and receiver. While the second training algorithm proposes an iterative training method similar to that proposed in [19]. Both of the proposed methods are trained and evaluated in AWGN, Rayleigh fading and a solid state high power amplifier (SSPA) channel and are compared with the W-GAN approach from [44] and the canonical AE [46]. In each channel, the proposed pre-training approach trains a transmitter and receiver which have a similar symbol error rate (SER) to the canonical AE and outperforms the W-GAN method [46]. The reduced sampling method is also compared with the DDPM approach, and an increasing number of iterations are shown to improve accuracy. However, diffusion models introduce additional complexity during the training cycles when used as a channel proxy, models such as GAN and VAE are advantageous in that they can approximate the channel distribution with a single forward pass during the E2E training phase.

Collaborative agent learning considers the OAL problem from a more general perspective, where more than one wireless communications system may participate in the learning process to develop a shared coded modulation scheme. Having multiple communicating agents introduces the need for a distributed coordination of the learning process and does not assume that each agent is comprised of the same machine learning model. However the focus remains the same as the other two methods, in that its goal is to learn a transmitter and receiver that are able to communicate optimally with respect to the channel environment.

Two agents collaborate to learn a coded modulation for fixed preambles in [47]. Both agents consist of a transmitter and receiver, where the transmitter is implemented by a NN and the receiver learns to classify symbols via a k-nearest neighbours (kNN) method using the known preamble as a reference. The transmitter updates its gradients using PG and by calculating the loss on the echoed estimate of the original message from the collaborating agent's receiver. The agents are trained in an AWGN channel environment and as the SNR decreases the learnt modulation is similar to QPSK, while as the SNR increases the resulting constellation resembles that of 16-QAM [47]. This demonstrates the ability of the agent to leverage the distance between constellation symbols and maximise throughput for changes in the channel environment [47]. The resulting BER for preamble lengths of 128, 256 and 512 does not perform as well as 16-QAM with the greatest loss of performance demonstrated for the shortest preamble [47]. The method relies on both agents sharing knowledge of a fixed preamble at each iteration of the learning cycle, which would either require a shared random seed, or an additional reliable channel for sharing information during learning. Additional channel usage is introduced as each learning cycle requires not only the transmission of the learnt modulation symbols from the transmitter but also estimates from the receiver of each collaborating agent. Coordination between agents also adds complexity however the approach demonstrates that it is possible to train heterogeneous systems to develop a common coded modulation.

The learning procedure is shown to improve as the amount of shared information increases between agents in [48]. Shared information in the DL approach to E2E and OAL takes the form of network design of transmitter and receiver, back-propagation of gradients, use of shared preambles or pilots, and feedback of receiver loss. In [48], the aim is to demonstrate a learning procedure with minimal shared information by extending on the collaborative protocol introduced in [47] and introducing a private preamble instead of shared preamble [48]. Varying degrees of information sharing are developed and evaluation in AWGN compared to the BER for QPSK demonstrate close performance to the baseline [48]. An additional set of models are developed in an SDR environment with performance indicated to be similar to the QPSK baseline [48]. The approach moves away from the requirement for a shared preamble however the complexity of the echo learning protocol remains, as coordination is required to realise cooperative learning. While the Collaborative agent learning is interesting especially in the ability to develop different kinds of collaborating agents, the other methods of Gradient approximation and Channel approximation produce more competitive results in comparison to conventional coding and modulation.

2.5 Adaptive Modulation and Coding

Training OAL enables a form of adaptation for DL models which is realised through tuning both transmitter and receiver on the true channel environment. However the process of training is time consuming and computationally demanding to develop a new code. Conventional AMC schemes achieve adaptation using a pre-designed series of coded modulations with differing spectral efficiency mapped to varying channel conditions. A coded modulation is selected based on the expected performance for current measurements of the channel environment, such as the estimated SNR. Such methods have been shown to outperform other approaches under varying channel conditions [14, 15]. The application of DL to AMC presents an opportunity for two approaches, the first is in using DL, as a method of parameter estimation and mapping to more accurately estimate performance of a given coding and/or modulation scheme for the current channel environment, and the second in developing new methods to support multiple coded modulation schemes for DL based communications systems.

In an OFDM system, the channel state must be estimated for multiple subcarriers, however it is difficult to estimate an optimal modulation and coding scheme to maximise performance in each subcarrier, hence a combined estimate of effective signal-to-interference and noise ratios (SINR) is applied in choosing an AMC [49]. The process of estimating the effective SINR is highly non-linear, involves a high dimensional number of environmental parameters and has no closed form to enable an analytic expression [49]. Therefore, a DL approach is proposed in [49] for estimation and mapping of AMC to maximise throughput. Two approaches are proposed for this purpose, the first develops a classifier base model to provide the mapping decision, the second integrates a genetic algorithm (GA) and DL to select the AMC scheme given channel measurement feedback [49]. In each case the DL and DL-GA methods outperform the conventional method in terms of throughput under a fading channel with delayed feedback [49]. Inputs to the model are provided by conventionally estimated CSI, however, a Channel approximation method may enable the method to learn directly from channel outputs rather than an intermediate step. The proposed method leverages DL as a decision-making approach for AMC, whereas it may be possible for an E2E approach to learn an appropriate modulation for the current channel conditions.

The received channel signal is used more directly where its power spectral density (PSD) is provided to a feed forward NN to estimate the SNR for AMC selection in [50]. A set of modulations including QPSK, 16-QAM and 64-QAM with varying code-rates are selected in the transmitter based on a mapping between SNR threshold and corresponding modulation and code-rate [50]. The system is compared with a conventional SNR estimation method and simulated over an OFDM system in a frequency-selective fading channel including Doppler frequency perturbations [50]. The PSD transformation for the received signal was chosen due to it being unaffected by phase and frequency perturbations, while the accuracy for the conventional method was shown to be impacted by changes in frequency [50]. Due to the relative independence on frequency, the proposed method demonstrated higher throughput and estimation accuracy for SNR under frequency distortion and was equivalent to ideal AMC where the SNR could be perfectly estimated [50]. The transformation of the received signal using the power spectrum enabled the NN to be unaffected by additional channel distortions and provide highly accurate parameter estimation for SNR. The role of the DL model focused on parameter estimation only, and did not perform mapping between SNR and a selected AMC scheme or examine how an AMC scheme could be learnt in an E2E manner given the channel environment.

The mixture of parameter estimation and control algorithm for selection of AMC scheme and transmit power, is a candidate for the application of RL. Space communications are subject to dynamic channel environments, and adopt both power control and AMC to provide reliable communications [51]. A Deep Q-Network (DQN) based approach is applied in [51] to fuse multiple inputs and define transmitter parameters to maximise reliability of communications between a geosynchronous equatorial orbit (GEO) satellite and ground station. The advantage of adopting a DQN approach was in the ability of a NN architecture to fuse multiple input sources and learn a mapping to achieve set communication objectives [51]. The developed sensing and control algorithm was demonstrated to outperform conventional AMC [51]. This type of application does not perform parameter estimation for the received signal, instead it leverages the DL to learn the optimal mapping between the observed state of the communications system, which includes estimation of CSI and map to an optimal choice of modulation scheme and order and transmit power [51].

The problem of estimating channel parameters and learning a modulation via E2E training is investigated in [52]. The channel gain for a simulated fading channel environment is estimated based on the received signal and subdivided into a number of intervals where an AE is trained, for each interval, with the E2E approach [52]. During operation the proposed method applies a blind channel gain estimation at the receiver which is sent to the transmitter over a feedback channel [52]. Once the estimated channel gain reaches the bounds for a new interval, the system then switches transmitter and receiver to the corresponding AE for that new interval [52]. The resulting system is demonstrated in comparison to the Hamming$(7,4)$ code and shown to achieve gains over the baseline method, additionally comparison is made between varying choices for the number intervals, and it is indicated that it is not necessary to use more than 8 intervals [52]. Gains are demonstrated in comparison to Receiver Tuning of a single AE, indicating that offline AMC has the potential to outperform OAL methods [52]. The approach estimated the channel gain for an assumed simulated channel via an analytic method, which may have instead applied a learning method for parameter estimation and mapping to AMC scheme. The proposed approach also trained multiple AE for a series of channel gain intervals at the same code rate, whereas training and model complexity could be addressed through the alteration of the DL architecture and training algorithm to support the learning of multiple modulations and code rates with a single model.

2.6 Summary

Synchronisation parameter estimation often designs the estimators in isolation from the communications system, although this is based on a data driven design ([24,25]). The application of parameter estimation also does not often consider performance on different kinds of modulations ([22,23,25]); the structure of preambles exhibiting strong correlation properties ([22,23,29]); or preambles with varying dynamic properties such as Chirps and Chaotic sequences ([21]). Hence, there is an opportunity to investigate the ability of DL to estimate synchronisation parameters for multiple modulations as well as both structured and randomised communication preambles.

Methods of E2E and OAL have demonstrated the ability of DL to learn modulations given either a simulated or observed channel environment. The challenges of E2E learning are related to the dimensionality of the message size and desired code rate ([3,32]); training and evaluation of DL transmitter and receiver in simulated channel environments with realistic channel perturbations ([34,35]); and the challenge of estimating channel features for synchronisation with reduced overheads introduced by preambles or pilot symbols ([13,36]). As well as facing similar challenges, training algorithms for the OAL environment are also challenged by the complexity of training algorithms. Such as training methods that have increased channel usage and assume feedback channels ([12,32,38,41]); require redundant models during training, such as discriminators in GAN based channel models ([13,19,44]); introduce complexity in the model due to an iterative training and generation process ([46]); or require coordination protocols which increase channel usage during training ([47,48]). Therefore there is opportunity to investigate how the training algorithm and DL architecture can be simplified to reduce the number of required transmissions, reduce the reliance on feedback, and eliminate the requirement for complex or redundant models used during training.

Finally, AMC presents a scenario where the choice between multiple coded modulation schemes can outperform a single conventional code over a dynamic channel environment. Much of the focus in the literature has been on the use of DL for channel parameter estimation ([49,50]), or has used conventional features combined with a learning method to map observed channel state to a selection of coded modulation ([51]). As demonstrated in the OAL literature, Channel Approximation can accurately estimate the channel distribution and may therefore provide opportunity for improved integration between parameter estimation and use of DL for mapping. Furthermore, the E2E learning of multiple code rates has so far required training multiple AE ([52]). An E2E architecture has yet to be developed to support multiple code rates with a single model, although there is some evidence that an estimated SNR input can assist the model to learn a different modulation. Hence there is further opportunity to develop both an E2E architecture and training algorithm to support learning of multiple code rates for use in AMC.

CHAPTER 3: DATA GENERATION AND SIMULATION

3.1 Introduction

The experimental methods and all relevant model simulations described in this doctoral thesis, make use of realistic data generation and simulation techniques during the training and evaluation of the proposed methods. This work is not undertaken on a physical hardware, therefore the primary purpose of simulation in this setting is to simulate the channel environment and the perturbations associated with that channel environment.

Since the primary function of the wireless communications system is to transmit and recover messages over the physical (simulated) channel, the messages that are transmitted constitute the source data. In this work, messages are binary sequences, or in the case of some preambles, a continuous sequence of values generated by a function, such as a chaotic map.

This chapter is divided into two sections, the first is dedicated to the data generation process for messages and the second describes the simulated channel environments and the associated perturbations which are applied to the output of the transmitter.

3.2 Data Generation

As described in Chapter 1 the task of a wireless communications system is to send a message M through a channel in such a way as to enable its recovery at the receiver. In this setting, when transferring information, the messages consist of a sequence of bits where each bit is either a 0 or a 1, $m \in [0,1]$. Given a message of K bits, the total possible number of messages (for the two possible values of each bit) is 2^K. In this work, K is also treated as the block size for the message. Except where otherwise stated (such as when using a fixed preamble), during training and evaluation the sequence of bits is random, without an assumed byte order. Hence, it is possible to supply an endless stream of bits for a given block size K during training and evaluation. A random seed (the constant 42 [53]), is defined to enable repeatable results.

Modulation is a process where the incoming bit stream is converted into a set of complex numbers or symbols which are then converted into a waveform. In the publications presented in this thesis the main forms of modulation are BPSK and QPSK. The BPSK modulation consists of two symbols, that is, the modulation order is 2, it has two symbols $u \in [-1+0j, 1+0j]$ and each symbol represents 1 bit. For each message bit m_t at time t, the corresponding BPSK complex symbol u_t is either -1 or 1 with a 0 complex component.

QPSK divides the constellation space into four possible symbols, for example: $u \in [-1-1j, -1+1j, 1+1j, 1-1j]$. It has a modulation order of 4 and each symbol represents 2 bits. The message block length K, must be even in this case. The in-phase (I) component of each symbol represents the amplitude and the quadrature (Q) component represents the phase. The modulation is converted into a waveform through multiplication with the exponential (Equation 3.1).

$$s(t) = u(t)e^{j2\pi f_c t} \tag{3.1}$$

Where f_c represents the carrier frequency Hertz.

For the work presented in Chapter 4, the modulation symbol is assumed to derived from the waveform $s(t) \in \mathbb{C}$, for the work presented in Chapters 5, 6 and 7, the modulation symbol is assumed to be the discrete complex value $u(t) \in \mathbb{C}$. In addition Chapter 4, incorporates a root raised cosine filter [54] to produce the resulting $s(t)$ waveform to prevent ISI.

Chapter 4 examines the use of fixed length binary and continuous preambles consisting of either the Barker 13 code and chaotic sequences generated from a predefined set of initial conditions. For random preambles, random bit sequences are used for the binary preamble and initial conditions are perturbed by random increments for the chaotic sequences. Both types of preambles result in a complex valued symbol $u(t) \in \mathbb{C}$ for each timestep t. The chaotic map functions are described in more detail in Chapter 4.

Prior to the channel the resulting output of the transmitter $z(t)$ is normalised to unit energy such that $||x||_2^2 \leq 1$, where x is the resulting modulation symbols (Equation 3.2).

$$z(t) = \frac{x(t)}{\sqrt{\sum_{i=1}^{L} x(i)^2 / L}} \tag{3.2}$$

For Chapters 5, 6 and 7 input to the modelling process consists of one-hot encoding of each unique message in 2^K possible messages. The comparative methods in those chapters perform evaluation leveraging conventional modulation and coding. In Chapter 4, inputs to the modelling process consist of outputs from the simulated channel and accompanying CFO perturbation.

3.3 Simulated Channel Environments and Perturbations

The channel environment is modelled as a transfer function $r(t) = h(z(t))$ and applies perturbations to the transmitter output. The publications contained in this doctoral thesis make use of AWGN, Rayleigh fading, Rician fading and SSPA to simulate amplification prior to the AWGN channel. Each of these channel environments are simulated with the basic form of Equation 3.3, where $n(t)$ represents additive Gaussian noise at time t, and $a(t)$ is a fading coefficient for time t. In AWGN, $a(t) = 1$ however, in Rayleigh fading and Rician fading the way it is calculated differs.

$$r(t) = a(t)z(t) + n(t) \tag{3.3}$$

In Rayleigh fading, the fading coeffient is drawn from a complex standard normal distribution $a \sim \mathcal{CN}(0,1)$. This fading coefficient is applied either per block, to simulate slow fading, or per bit, to simulate fast fading, the usage is stipulated in each publication. In Rician fading, the normal distribution is parameterised with $\mu = \sqrt{K/(2(K+1))}$ and $\sigma = \sqrt{1/(2(K+1))}$ where K is a constant Rician factor. When K is small the Rician channel behaves similarly to the Rayleigh fading channel, and as K increases the Rician channel is closer to AWGN. The value chosen for the Rician factor K is described in each publication where Rician fading is used (Chapter 6 and Chapter 7).

Chapters 6 and 7 provide an additional channel featuring amplification at the transmitter. The amplifier is based on the Rapp model [55] shown in Equation 3.4, with parameters for the limiting output amplitude A_0, amplifier gain ν and smoothness p. The values assigned these parameters are specified in the corresponding publications for each chapter.

$$g(A) = \nu \frac{A}{\left(1 + \left[\left(\frac{\nu A}{A_0}\right)^2\right]^p\right)^{1/2p}} \tag{3.4}$$

The SSPA is applied to the output of the transmitter $z(t)$ (Equation 3.5) and the resulting signal $z'(t)$ is then fed into the channel transfer function.

$$z'(t) = g(|z(t)|)e^{j\angle z(t)} \tag{3.5}$$

All channels include an additive Gaussian noise term $n(t)$, which is simulated for the desired level of SNR dB. The ratio of energy per information bit to noise power spectral density E_b/N_0 dB, is specified during the simulation, and ranges between 0 to 15 dB. A code rate $R = K/N$ is applied to convert to the ratio of energy per symbol to noise power spectral density E_s/N_0 dB $=$ E_b/N_0 dB $+ 10\log_{10}R$. The result is converted to linear form $E_s/N_0 = 10^{E_s/N_0 \text{ dB}/10}$ and the signal is used to estimate the energy per symbol $E_s = \sum_{t=1}^{L} z(t)^2/L$ where L is the length of the signal. The resulting noise power spectral density is estimated as $N_0 = E_s/(E_s/N_0)$ to determine the variance $\sigma^2 = N_0/2$ for drawing from the complex normal distribution $n(t) \sim \mathcal{CN}(0, \sigma^2)$.

Chapter 4 is concerned with results of the simulation of CFO. This research adds a random perturbation for the frequency f_{offset} selected from a random uniform distribution between ± 5 kHz. The resulting frequency is then applied to the output of the transmitter replacing the f_c term in Equation 3.1.

CHAPTER 4: PAPER 1 - USING SEQUENCE-TO-SEQUENCE MODELS FOR CARRIER FREQUENCY OFFSET ESTIMATION OF SHORT MESSAGES AND CHAOTIC MAPS

4.1 Introduction

This chapter presents a copy of the article published in IEEE Access (vol. 10 (2022), pp 119814-119825, ISSN 2169-3536, doi:10.1109/ACCESS.2022.3221762. `https://ieeexplore.ieee.org/document/9947074`).

This work examines the task of Synchronisation parameter estimation for blind modulations on short preambles. The preambles include two conventional modulations, BPSK and QPSK as well as three continuous preambles based on Chaotic maps, the Circular, Quadratic and Zadoff-Chu maps. An architecture for CFO estimation is developed which performs both coarse and fine grained estimation. Two approaches are compared, one based on a discretised fine estimate and the second which performs the fine estimate with a regression-based approach. This article is motivated due to the ability of DL to provide accurate estimation with limited data in comparison to conventional FFT based approaches, and for the ability of DL to generalise over multiple modulations. CFO estimation has applications not only to synchronisation, but also in application of modulation classification, and the ability of DL to perform blind estimation on limited data is a significant capability which reduces energy use and delay in correcting received signals.

Research Highlights

- A branching DL architecture is developed for the multi-stage estimation of CFO for conventional as well as chaotic modulations.

- The proposed approach is demonstrated to outperform several FFT based methods for

CFO on short preamble lengths.

- The proposed approach is also demonstrated to outperform correlation based methods with limited upsampling. However, correlation methods with a higher upsampling factor are demonstrated to be more accurate at the expense of execution time.
- The article evaluates the proposed approach both on fixed and randomised preambles, demonstrating the ability to generalise over multiple sequences.
- A comparison between model performance with and without feature engineering is presented, as well as an analysis of the feature importance between two variations of the proposed model.

4.2 Published Article 1

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 26 October 2022, accepted 8 November 2022, date of publication 14 November 2022, date of current version 17 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3221762

RESEARCH ARTICLE

Using Sequence-to-Sequence Models for Carrier Frequency Offset Estimation of Short Messages and Chaotic Maps

CHRISTOPHER P. DAVEY[1], (Graduate Student Member, IEEE), ISMAIL SHAKEEL[1,2], (Senior Member, IEEE), RAVINESH C. DEO[1], (Senior Member, IEEE), SANCHO SALCEDO-SANZ[3], AND JEFFREY SOAR[4]

[1]School of Mathematics, Physics and Computing, University of Southern Queensland (USQ), Springfield, QLD 4300, Australia
[2]Information Sciences Division, Defence Science and Technology Group (DSTG), Contested Communications Branch, Canberra, SA 5111, Australia
[3]Department of Signal Processing and Communications, Universidad de Alcalá, 28805 Alcalá de Henares, Spain
[4]School of Business, University of Southern Queensland, Springfield, QLD 4300, Australia

Corresponding authors: Christopher P. Davey (christopher.davey@usq.edu.au) and Ravinesh C. Deo (ravinesh.deo@usq.edu.au)

This work was supported in part by the Department of Defence of the Commonwealth of Australia under Defence Science Partnership Scholarship under Agreement 10254.

ABSTRACT Deep Learning methods have produced good carrier frequency offset estimations for short message sequences in comparison with methods based on the Fast Fourier Transform. However, these performance gains were observed for short ranges of frequency offsets, sequences with predefined pilot symbols and periodic modulation schemes. Chaotic modulation has an advantage over periodic signals in offering security through the continuous changes produced by parameterising the chaotic map function. However, synchronisation of chaotic map parameters in coherent receivers is dependent on the carrier recovery of phase and frequency which dramatically reduces the demodulation performance under high noise levels. This article presents a stacked sequence-to-sequence neural network architecture for blind carrier frequency offset estimation of both periodic and chaotic modulation schemes. The results obtained demonstrate better performance than conventional methods in low SNR for the Additive White Gaussian Noise channel. While this technique operates without feature engineering, the results demonstrate that data augmentation produces a higher degree of accuracy for such models, indicating the benefit of integration with conventional signal pre-processing steps as part of the deep learning pipeline. The proposed neural network architecture is shown to perform carrier frequency offset estimation, not only for the selected periodic modulations, but also in the case of highly non-linear chaotic maps. This suggests the applicability of deep learning methods for synchronisation in waveforms that employ chaotic modulation schemes for secure communication and for applications where short and sporadic messaging are required (e.g., Internet of Things).

INDEX TERMS Chaotic communication, deep learning, fast fourier transforms, frequency synchronisation, carrier frequency offset estimation.

I. INTRODUCTION

The accuracy of Carrier Frequency Offset (CFO) estimation methods based on the Fast Fourier Transform (FFT) in single carrier communications is dependent on the sample length of the message, and on the Signal to Noise Ratio (SNR) [1]. Short sample message lengths are advantageous in low power Internet of Things (IoT) applications and pilot signals used for signal detection and synchronisation. Deep Learning (DL) methods have demonstrated to outperform FFT-based

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Wang.

methods under similar constraints [2], [3]. However, much of the experimentation to date has focused largely on phase amplitude modulation (PAM) or M-ary phase shift keying (M-PSK) modulations, and has not investigated the potential application to chaotic modulation techniques.

Chaotic modulations present a method for providing physical layer security, and are well suited to address the constraints placed on IoT applications [4]. Due to the continuously changing signal which results from parameterisation of the chaotic map sequence, chaotic modulations exhibit high autocorrelation for the same symbol and low cross-correlation between symbols [5]. This characteristic is advantageous for coherent detection, where each symbol is correlated with a potential mapping function at the receiver and is resilient to small levels of noise [5]. However to achieve demodulation the receiver is required to estimate the parameters for each chaotic map function which is known as sequence synchronisation [6]. Sequence synchronisation for chaotic maps is dependent on accurate estimation and removal of the CFO [6], [7]. For estimating frequency offsets in chaotic maps, autocorrelation methods are shown to be effective for fixed preambles [8], however these methods are difficult to implement for variable and non-repetitive sequences.

Given that deep neural networks can learn non-linear features, the estimation of CFO for randomised chaotic sequences is an application well suited to such methods. In this article we propose a data driven method for the estimation of the CFO in short sequences of BPSK, QPSK modulations, as well as for the Circular, Quadratic and Zadoff-Chu chaotic maps. The approach is applied to both fixed preamble and randomised sequences. The model performs an iterative estimation of the frequency offset using a sequence-to-sequence (Seq2Seq) block at each level. This approach is capable of more accurate CFO estimation for the M-PSK modulations in comparison with the FFT and Phase Locked Loop (PLL) approach. While brute force cross-correlation is more accurate without down-sampling at the matched filter (at the expense of execution time), the DL method is more accurate when compared with cross-correlation on the shorter down-sampled signal. The network can produce CFO estimates directly from the In-phase and Quadrature (IQ) values of the received signal, however data augmentation is shown to provide an advantage for the accuracy of the estimation.

A. BACKGROUND AND RELATED WORK

The use of the FFT is demonstrated to perform an approximation for the maximum-likelihood function of the parameters in a sinusoidal signal corrupted by Gaussian noise in [9]. The length of the FFT determines the accuracy of the measurement, and was found to be optimal at up to 4 times the length of the signal [9]. As the frequency step size of the FFT produces a coarse estimation, an interpolation is required to produce a finer estimate. In the case of [9] an iterative secant method is applied to the fine estimate of the frequency but is indicated to produce a larger error in low SNR [9]. The threshold for the variance of the estimator in [9] is shown to be optimal above an SNR between 15 dB and 17 dB in [10] for corresponding sequence lengths between $N = 64$ to $N = 2048$.

Interpolation methods using points either side of the maximum value for the FFT are applied to calculate an adjustment term for the frequency estimate in [11] and [12] and improve on the method in [9]. These methods are shown to have a bias for short sequences and low SNR in [10] which proposes three and five point interpolation methods making use of the phase information in the FFT coefficients. Several methods of interpolation are compared in [13] which also makes use of three coefficients to demonstrate a method that approaches uniform error variance above 2 dB. An extended number of fourier coefficients weighted by an approximation of their mean square error are combined to estimate the frequency offset in [14], resulting in an estimator approaching the lower bound of variance close to 5 dB. However each of these methods share limitations in lower SNR and for short sequences. In addition the application of the FFT is applicable for periodic signals and are not appropriate for use with those chaotic modulations which do not exhibit distinctive peaks within the power spectrum.

DL approaches, in particular convolutional neural networks (CNN), are demonstrated to outperform FFT based methods on estimation of CFO for short random sequences in 1-bit ADC's at low SNR in [2]. The selection of DL models is able to extrapolate well over a wider range of SNR (between -20 and 40 dB), even though they are trained on a subset of the SNR (between 0 and 10 dB) [2]. The 1-bit quantization method reduces the amount of information available to the network for training [2] and for conventional methods it is known to require up to four times oversampling for the estimation of offset parameters [15]. In conventional methods, knowledge of modulation order M is applied to remove the modulation from the signal prior to the application of FFT estimation, however the generality of the 1-bit ADC in [2] did not motivate an exploration of the impact of the modulation on CFO estimation. As our method is applied after down-sampling at the matched filter output, the type of modulation is shown to have an influence on estimation accuracy for both FFT and DL approaches.

Further indication that DL can provide good frequency offset estimation for sinusoidal waveforms in low SNR is described in [3]. The network architecture was constrained specifically to the fully connected network (FCN) with the number of input nodes representing the length of the signal to be processed and being dependent on the range of the frequency offset, requiring larger dimensions for wider ranges of frequency [3]. FCN networks require a larger number of connections between layers as opposed to the CNN [16], hence consideration of CNN layers would provide flexibility for processing multiple signal lengths with a constant number of layer parameters. Although the choice of network architecture limited the range of frequency offset, it was shown that the FFT and DL methods did decrease in accuracy under shorter signal lengths [3]. To address a wider frequency offset range, as well as several modulations, this article proposes the stacked network architecture, which incorporates CNN layers to extract features at each level rather than fully connected layers.

Short signals prevent the FFT from accurate spectral estimation due to the resulting coarse resolution, whereas a DL method for super-resolution estimation of the approximate spectrogram is proposed in [17]. A combination of both FCN (linear) and CNN layers are applied in the architecture, taking advantage of the ability of the CNN to accept multiple resolutions of input during training to learn translation invariant features [17]. A customised minimum distance loss is applied during the learning procedure and the model is shown to produce more accurate estimation than the periodogram and eigenvector (MUSIC) based estimators at a limited range of SNR [17]. The model is trained and tested on the complex sinusoid with amplitudes, frequency and phase selected from random normal distribution at different parameters [17]. A fixed output resolution is used to estimate the pseudo-spectrum of the signal which is then mapped onto a known frequency range [17], the resolution is dependent on the signal length and is fixed. Our proposed stacked model refines the peak frequency estimate at increasing resolutions for each stack in the network and estimates an error correction term to produce a high resolution estimate for the carrier frequency offset at the final layer.

The CNN is leveraged in the literature on the CFO estimation task, however as the signal varies over time, a recurrent neural network (RNN) may be applied to learn time dependent features over the signal. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) network models are trained to perform CFO estimation with the short training field (STF) of the IEEE 802.11ah preamble frame in [18]. Results demonstrate that the network performs well on the CFO estimation task in comparison with the conventional correlation method in low SNR [18]. The STF is a fixed pattern within the frame and is useful in simplifying the process of timing and CFO estimation [18]. It is designed to improve the resulting accuracy of the estimation method. In the proposed method, we experiment with both the fixed preamble as well as randomised sequences for several modulations and demonstrate that the DL approach can learn to estimate the CFO even where the modulation exhibits chaotic behaviour. In the proposed architecture, recurrent LSTM layers learn time dependencies resulting from features modelled by CNN layers and are organised in encoder-decoder blocks which share the hidden state for learnt time dependencies between them.

A common element in the cited literature is that the DL method is more accurate than conventional methods in low SNR and for short sequences. While the FCN layer is applied in [3] due to the constraints of the experiment, the CNN has advantages as an effective choice for feature extraction in the CFO estimation task [2], [17] and the use of the LSTM is shown to be effective in [18]. It is clear a DL model can be constructed for a single modulation, the impact of estimating CFO for multiple modulations has not been investigated for such an approach. Spectral methods are optimal under the right conditions and would be useful to incorporate into the design of the network model as demonstrated in [17]. The chaotic map becomes deterministic when the state parameters are known. A recurrent network modelling approach may demonstrate the ability to learn implicit information from the signal, thereby aiding estimation of the CFO. A combination of RNN and CNN would enable a DL model to both extract translation invariant features as well as learn time dependent features. This article proposes a stacked architecture which estimates the probability of the peak frequency as well as an error correction term using sequence-to-sequence blocks comprised of CNN and LSTM units.

The rest of the paper has been structured in the following way: The next section describes the system model, as well as the conventional carrier offset estimation method. It also explains the proposed model architecture, as well as the data augmentation applied when training the model. Section III shows the experimental results obtained when the proposed DL approach is applied to a number of CFO estimation tasks. A discussion on these results is also provided in this section. Section IV closes the paper, by giving some final concluding remarks on the research carried out.

II. METHODS

When transmitted over a channel, the baseband signal $s(t)$ is subject to perturbations of timing t, phase θ and carrier frequency f_0 offsets, shown in Equation (1), where $a(t)$ represents the signal modulation after filtering, and $n(t)$ represents Additive White Gaussian Noise (AWGN).

$$s(t) = a(t)e^{j\theta}e^{j2\pi f_0 t} + n(t) \tag{1}$$

In this work the proposed model is trained on several modulations, which include Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), as well as chaotic Circular, Quadratic and Zadoff-Chu maps. Frequency offsets for M-PSK modulations are estimated in two stages: first, a coarse estimate $\hat{f}_1$ is given by the position of maximum frequency of the coarse grained FFT (Equations (2)-(4)). The derivation for the use of the Discrete Fourier Transform (applied through the FFT) as an approximation for the maximum-likelihood estimator of the CFO is described in Rife and Boorstyn [9], in this article we apply the Matlab coarse frequency estimator [19] which is derived from the use of the FFT in [20]. The received signal $s(t)$ is first raised to the Mth power $z(t) = s(t)^M$, then the FFT is calculated giving $S(k)$ (Equation (2)). The index k_m of the frequency, having the maximum absolute value for $S(k)$ (Equation (3)) is then divided by the modulation order M ($M = 2$ in BPSK and $M = 4$ in QPSK) and is scaled by the sampling frequency f_s over the length of the FFT N (Equation (4)). After the coarse estimate, a fine frequency adjustment $\hat{f}_2$ is estimated via a PLL implemented by the Matlab carrier synchronisation function [21] derived in [22]. The difference in phase error estimates $\Delta\theta$ produced by the PLL are scaled to the frequency estimate via the sampling rate f_s and the down-sampling rate d, and the operation is averaged to estimate the adjustment for the frequency offset (Equation (5)). Finally, the frequency offset is estimated as the sum of the coarse frequency estimate and the fine frequency adjustment (Equation (6)). Improvement in accuracy can be gained by increasing the resolution of the FFT, results from [9] recommend a resolution up to four times the length of the original signal, depending on performance constraints. In our experiments the FFT resolution is set to $4\times$ the down-sampled received signal of 104 samples.

Two FFT interpolation methods are employed for comparison. Both methods adjust the index k_m through an estimate of the difference to the peak of the FFT, $\hat{\delta}$ and add it to the index as in Equation (10), the updated index, k_{adj} is then applied in estimating the frequency f_1 (replacing k_m with the adjusted index k_{adj}). The first interpolation method is described in [13] where the two values either side of the maximum index are used to estimate the difference from the peak of the FFT (Equation (7)), this method reduces the bias of the quadratic interpolation. The second method is proposed in [14] which incorporates all FFT coefficients (in $K < N/2 - 1$) and calculates an estimate for the adjustment $\hat{\delta}_k$ at each coefficient index k (Equation (8)). These estimates are aggregated through weighting each with an approximate of their mean square error term (Equation (9)) [14]. In the results section the first interpolation method is indicated on plots as 'Jacobsen' and the second 'Candan'. Both methods are suitable for use in multiple iterations, however in our comparison we generate results with only one application of each method.

$$S(k) = \sum_{n=0}^{N-1} z_i e^{j2\pi kn/N} \tag{2}$$

$$k_m = \text{argmax}|S(k)| \tag{3}$$

$$\hat{f}_1 = \frac{f_s}{N}\frac{k_m}{M} \tag{4}$$

$$\hat{f}_2 = \frac{1}{N}\sum_{i=1}^{N}\frac{f_s}{d}\frac{\Delta\theta}{2\pi} \tag{5}$$

$$\hat{f}_0 = \hat{f}_1 + \hat{f}_2 \tag{6}$$

$$\hat{\delta} = -\text{Re}\left[\frac{S_{k_m+1} - S_{k_m-1}}{2S_{k_m} - S_{k_m-1} - S_{k_m+1}}\right] \times \text{Re} \tag{7}$$

$$\hat{\delta}_k = \frac{N}{\pi}\tan^{-1}\left(\tan(\frac{\pi k}{N}) \times \left\{\frac{S_{k_m+k}e^{-j(\pi/N)k} - S_{k_m-k}e^{j(\pi/N)k}}{S_{k_m+k}e^{-j(\pi/N)k} + S_{k_m-k}e^{j(\pi/N)k} - \frac{2S_{k_m}}{\cos(k\pi/N)}}\right\}\right) \tag{8}$$

$$\hat{\delta} = \frac{\sum_{k=1}^{K} 1/\sin^2(\pi k/N)\hat{\delta}_k}{\sum_{k=1}^{K} 1/\sin^2(\pi k/N)}, K < N/2 - 1 \tag{9}$$

$$k_{adj} = k_m + \hat{\delta} \tag{10}$$

The cross-correlation method is applicable where a template such as a pilot signal is known. The template signal is rotated by frequency steps $f_1, f_2, \ldots, f_n$ between the range of the expected frequency offset (in our experiments ± 5 kHz). The complex cross-correlation between the received signal and the distorted template is calculated and the maximum cross-correlation is used to determine the index of the frequency estimate. In our randomised experiments, the DL model does not have any knowledge of the template used for the comparative method, whereas in the fixed preamble setting it is trained on a fixed sequence. Cross correlation is performed prior to down-sampling at $4\times$ sample length and post down-sampling at $2\times$ sample length for comparison. This method is computationally expensive and is most accurate on small frequency ranges and longer signal lengths.

A. DATA GENERATION

The data used in training and evaluation are divided into two experimental settings, the fixed preamble setting and the randomised sequence setting. In the fixed preamble setting, M-PSK sequences are generated by repeating a fixed message containing the 13 bit Barker code. For the chaotic maps, the initial conditions are predefined along with a fixed length for the recurrence relation within the map. Randomised sequences consist of random bits for the M-PSK messages and sliding windows of chaotic maps. Both types of sequences (fixed and random) are constructed where the bit sequence length is dependent on the number of bits per symbol and produce 2 samples per symbol resulting from matched filtering (up-sampled at $8\times$ and decimated at $4\times$ per sample respectively). After applying a root raised cosine matched filter at the transmitter and receiver, a 52-bit sequence for BPSK and 104-bit sequence for QPSK generate 104 samples. In the chaotic modulations 52 symbols are mapped to a resulting 104 symbols after matched filtering. All sequences are 104 samples in length.

Chaotic sequences cannot be randomised in the same manner as bit sequences, since they depend upon the initial conditions for each symbol and are parameterised depending on the mapping function. Given their reliance on successive feedback, a randomised chaotic sequence is generated by randomly selecting the number of feedback iterations from an initial condition and stepping the mapping function over the sequence length while storing the feedback signal to use as the initial conditions for the next sequence. The mapping functions for each of the chaotic maps are shown in Table 1, along with the feedback parameter and initial condition parameters. Figure 1 illustrates the IQ values for each of the corresponding map functions.

During the data generation process, no phase rotation is applied, and the frequency offset is selected from a random uniform distribution within the range ± 5 kHz with a sampling frequency $f_s = 1$ MHz. Noise is added for SNR, $E_s/N_0 = 0 \ldots 9$ dB with the noise variance σ^2 being estimated from parameters E_s and N_0 in Equations (11)-(13), where E_s is the energy per channel symbol, N_0 the noise

TABLE 1. The set of chaotic map functions and their initial parameters used in generating sliding sequences.

Map Function	Parameters	Initial Value
Circular Map $x_{i+1} = x_i + b + \frac{a}{2\pi}\sin(2\pi x_i)$	$0 \leq a, b \leq 1$ $a = 0.5$ $b = 0.2$	$x_0 = 0.7$
Quadratic Map $x_{i+1} = b - a x_i^2$	$a = 4$ $b = 0.5$	$x_0 = 0.15$
Zadoff-Chu Map $x_i = \exp\left(-j\frac{\pi u i(i+1)}{N}\right)$	$0 < u < N$ $\gcd(N, u) = 1$	$N = 63$ $u = 1$ $x_0 = 0.7$

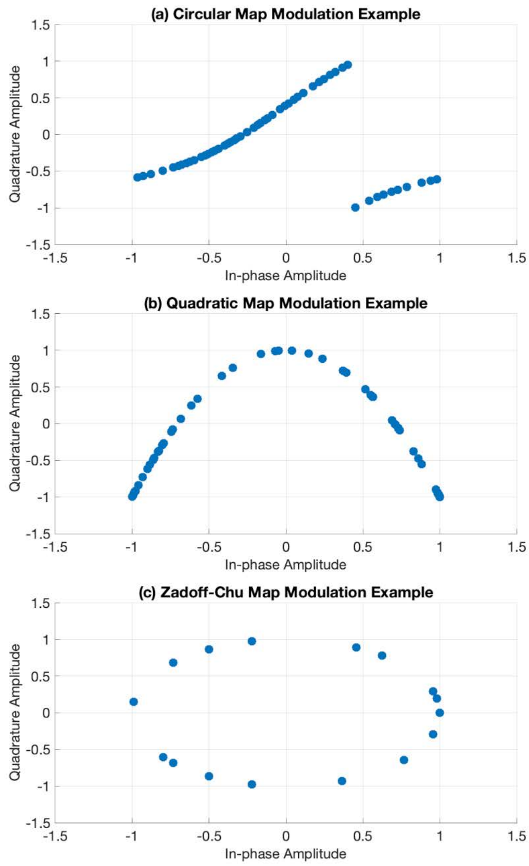


FIGURE 1. Example IQ plots of the chaotic map functions for a) Circular, b) Quadratic and c) Zadoff-Chu maps.

power spectral density, L the number of symbols, and n the bits per symbol. For training the network an offline dataset of 102400 sequences is generated for each modulation (502400 sequences) and each sequence is labelled with the corresponding random frequency offset that was applied to distort the signal.

$$E_s = \frac{\sum_{t=1}^{L} s(t)}{L/n} \tag{11}$$

$$N_0 = \frac{E_s}{E_s/N_0} \tag{12}$$

$$\sigma^2 = N_0/2 \tag{13}$$

B. NETWORK ARCHITECTURE

The intuition applied to the design of the network architecture was that the network should be capable of multiple stages of refinement in the task of frequency offset estimation. A stacked architecture was arrived at such that each stack would successively estimate a discrete set of steps for the frequency range where the step size decreases at each level in the stack. The final level then estimates the error between the coarse estimate of the previous layer and the target frequency. For comparison, the error adjustment layer is implemented with two approaches. The first applies a classification approach that is constrained within ± 100 Hz of the coarse estimate. The second approach applies a direct regression to provide a continuous error correction to compensate for broader variation of the error between the coarse estimate and target frequency offset.

Each stack consists of a subnetwork block which is responsible for learning features and performing estimation for that block. To perform feature extraction, as well as learn recurrence relationships, a sequence-to-sequence (Seq2Seq) network is defined within the feature extraction block. The Seq2Seq architecture follows the approach first defined in [23], however beam search is not applied during estimation and the inclusion of Convolutional layers differs from the original model. The block design includes a Convolutional (CNN) layer to extract input features, a bidirectional Long Short-Term Memory (LSTM) encoder, latent space implemented as a CNN layer, a bidirectional decoder LSTM layer followed by an output CNN layer. Classification is provided by a Dense block with a soft-max activation while regression is achieved with a tanh activation. Regularisation is provided by applying Batch Normalisation [24] following each CNN and intermediate Dense layer, and Layer Normalisation is applied after each LSTM layer. Max-pooling is applied to the output of intermediate CNN layers with Global Average Pooling applied prior to the Dense layer.

Aside from the estimation output, the hidden LSTM state is shared between encoder and decoder LSTM, and the hidden state of the decoder is forwarded to the encoder in the subsequent stack. The latent CNN state is also forwarded between network stacks and concatenated with the input features for the encoder in the subsequent stack. These skip connections enable multiple forward paths fusing latent features and sharing hidden recurrent state throughout the network and enable gradient flow during back-propagation [25]. Such connections are proposed to enable ensemble like behaviours in deep networks [26]. Figure 2 presents the schematic view of the sequence-to-sequence block as well as the dense estimator blocks for the network output and the interconnection between the blocks is illustrated in Figure 2. Three stacks were defined, with frequency bins of 100 and 50 Hz for both the classifier and regressor networks. A frequency adjustment of ± 100 Hz is applied for the final estimator of the classifier network, and a single continuous parameter applied in the final estimator of the regressor network. Table 2 lists the number of units for each layer type.

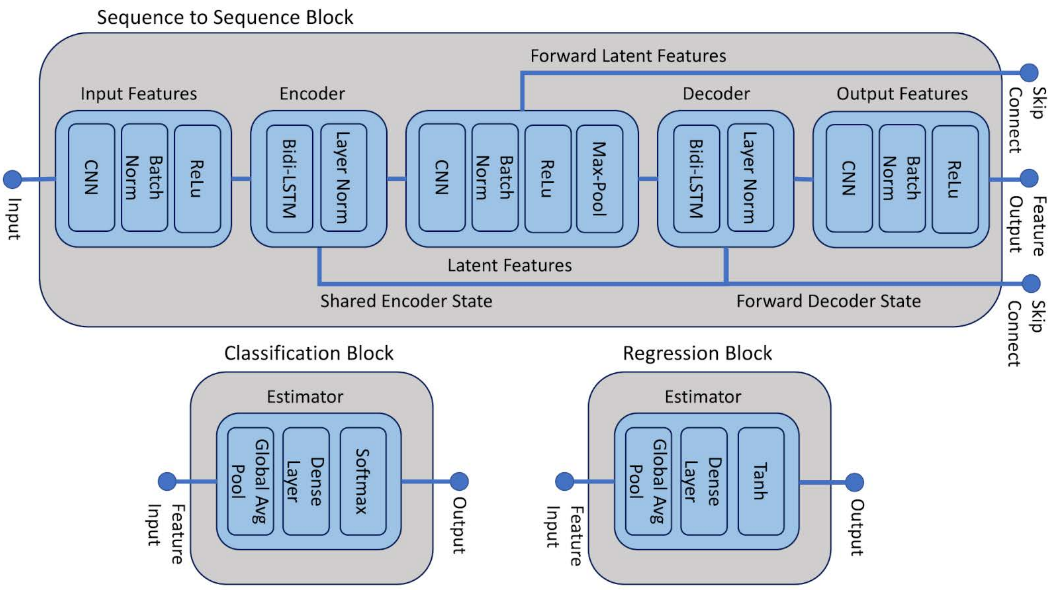


FIGURE 2. Sequence to Sequence blocks with CNN feature extraction are interconnected with paths for hidden recurrent state and latent CNN state. The final output of each block pools the output features of the sequence to sequence block to produce the estimate for either the frequency bin or the frequency error.

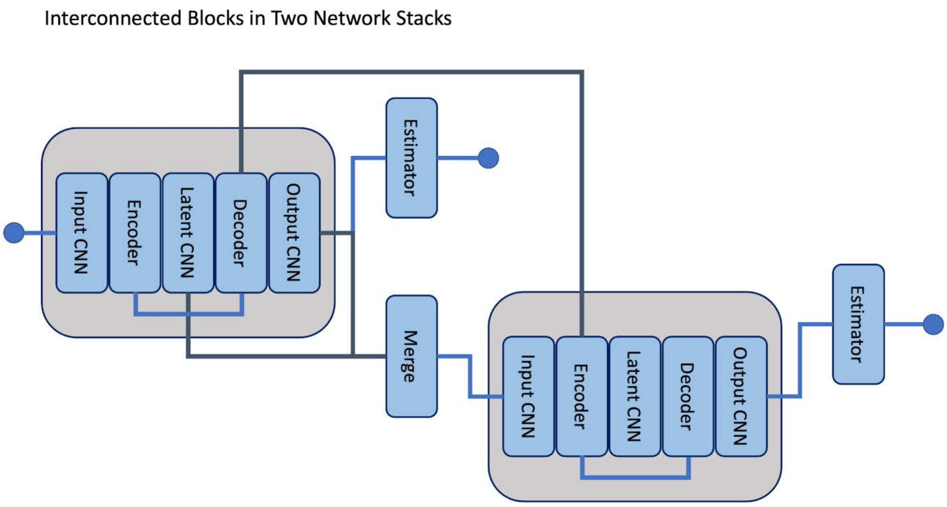


FIGURE 3. Interconnection between two sequence-to-sequence blocks shares the decoder hidden state with the encoder of the subsequent block and merges the latent CNN state with the CNN output via a concatenation.

TABLE 2. The set of chaotic map functions and their initial parameters used in generating sliding sequences.

Layer Type	Units/Filters
CNN	32
LSTM	128
Dense Classifier units (bin Hz) per stack	102 (step size 100 Hz) 202 (step size 50 Hz) 202 (step size 1 Hz)
Dense Regressor	1

During the network's training, the data set is partitioned into 50% training, 20% validation and 30% test. A cyclical learning rate schedule [27] was applied which allowed the learning rate to oscillate between 0.0001 and 0.001. Input data was scaled by dividing the input signal by the l^2-norm and min-max normalising with parameters ± 1. Target frequency is min-max normalised with parameters ± 5 kHz. Back-propagation is performed with Adam optimisation [28]. Cross-entropy loss is applied to the classification estimator and mean squared error loss is applied to the regression estimator. Each stack is trained iteratively, and the weights of each previous stack are frozen prior to training the subsequent stack. When training the final stack, the difference between the previous stack frequency estimate and the true target frequency is calculated and applied as the target after min-max normalisation (± 5 kHz).

TABLE 3. Data augmentation produced 17 features prior to the input for the network. Each of the features were derived from steps used in conventional synchronisation.

Operation	Description
$y_t = r_{t+1} - r_t$	Lag-1 Differences of IQ values of received signal r. The difference between lags are used in the fine tuning of coarse grained estimation in conventional methods.
$[r^2, r^4]$	Raised Powers of IQ values. In PAM and PSK modulations raising the signal to the order of the modulation results in a constant phase for the IQ coordinates.
$[\mathrm{FFT}(r^2), \mathrm{FFT}(r^4)]$	FFT for raised powers of received IQ values in r. A coarse grained FFT of the down-sampled signal is used to produce coarse estimates for the frequency.
$\hat{R}(k) = \frac{1}{N-k}\sum_{m=k}^{N-1} r(m)r^*(m-k)$	Autocorrelation of received signal r. The autocorrelation of the signal is used in conventional phase and frequency correction procedures.
$p = [\lvert r\rvert, \angle r\rvert]$ $[p^2, p^4]$	Raised Polar form p of received signal r. The raised polar form is used in conventional phase estimation algorithms.
$y'_t = r^*_{t+1} r_t$	Lag-1 Conjugate for received values in r. Conventional frequency estimation may employ the conjugate between two samples of a given interval (in this case lag 1).

The network models are trained under two experimental settings, fixed preambles and randomised sequences, with each setting producing separate models (eight individual models in total, four model variants in each setting). A third experiment explores the difference in training on a single modulation, as opposed to multiple modulations. In this task, two variants of the network model are independently trained on QPSK and Quadratic map modulations for each setting, resulting in eight individual models.

C. DATA AUGMENTATION

A comparison is made between models trained with and without data augmentation. For those networks that are trained without data augmentation, the complex signal is represented as a matrix with two columns for the in-phase and quadrature components. Those networks trained with data augmentation were supplied with 17 features derived from the treatment of the complex signal in conventional synchronisation algorithms, these are described in Table 3.

During evaluation, a separate feature importance analysis is undertaken by iteratively assigning uniform noise to each feature and calculating the difference in performance between the baseline model and the noisy input data.

III. RESULTS

The Mean Absolute Error (MAE), in Hz, produced by the Stacked Model and the FFT/PLL method for the CFO estimation task is shown in Figure 4 for BPSK and QPSK modulations between 0 and 15 dB SNR in both experimental settings. Accuracy differs on each modulation for both the proposed and conventional methods, with the proposed method achieving higher accuracy on short sequences at 104 samples than the FFT/PLL method with 4× FFT resolution. Similarly the MAE, in Hz, for each chaotic map sequence is shown in Figure 5, where the panels on the left hand side show the proposed stacked network results for estimation using 104 samples and those on the right showing the effect of sample length on the brute force correlation method at 2× and 4× sample lengths (208 and 416 samples). The stacked network is more accurate than the cross-correlation with 2× upsampling, however the cross-correlation at 4× upsampling demonstrates much higher accuracy at the expense of execution timing. Like the BPSK and QPSK modulations, the kind of chaotic map influences the accuracy of the estimate.

TABLE 4. Comparison between STACKNet$_C$ and STACKNet$_R$ on fixed preamble sequences indicates a minor difference between model variants when data augmentation is applied. A slight improvement in MAE Hz does result from the regression model in comparison to the classification model.

Comparison	Mean Difference in MAE Hz
$\mathrm{MAE}_{\mathrm{STACKNet}_C 17F} - \mathrm{MAE}_{\mathrm{STACKNet}_C}$	-0.45
$\mathrm{MAE}_{\mathrm{STACKNet}_R 17F} - \mathrm{MAE}_{\mathrm{STACKNet}_R}$	0.99
$\mathrm{MAE}_{\mathrm{STACKNet}_C} - \mathrm{MAE}_{\mathrm{STACKNet}_R}$	-4.49
$\mathrm{MAE}_{\mathrm{STACKNet}_C 17F} - \mathrm{MAE}_{\mathrm{STACKNet}_R 17F}$	-5.94

Comparison is made between two configurations of the network architecture where error adjustment is implemented with either a classification layer (STACKNet$_C$) or as a regression layer (STACKNet$_R$). In addition, models are trained with and without data augmentation as indicated by the postfix $17F$. In the fixed preamble setting, there is little difference between models that are trained with and without data augmentation, while the regression model achieves a lower MAE Hz on average than the classification model, indicated in Table 4. On randomised sequences, those models trained with data augmentation demonstrate slightly lower MAE (Hz) on most modulations and SNR. While the performance of the augmented classification and regression models (STACKNet$_C 17F$ and STACKNet$_R 17F$) are similar, the regression model does appear to perform better on most modulations for randomised sequences, especially on QPSK and Quadratic modulations which exhibit higher MAE (Hz) for all models. Table 5 shows the mean improvement in MAE (Hz) between those models in the random setting.

In a separate experiment, the model architecture with data augmentation is trained on single modulations for QPSK and Quadratic maps. Figure 6 indicates a lower MAE Hz for the regression model with the exception of random QPSK where performance between the two variants are close. Figures 4 and 5, indicate that training on a single modulation produces results similar to training on multiple modulations and that performance is dependent on the type of modulation.

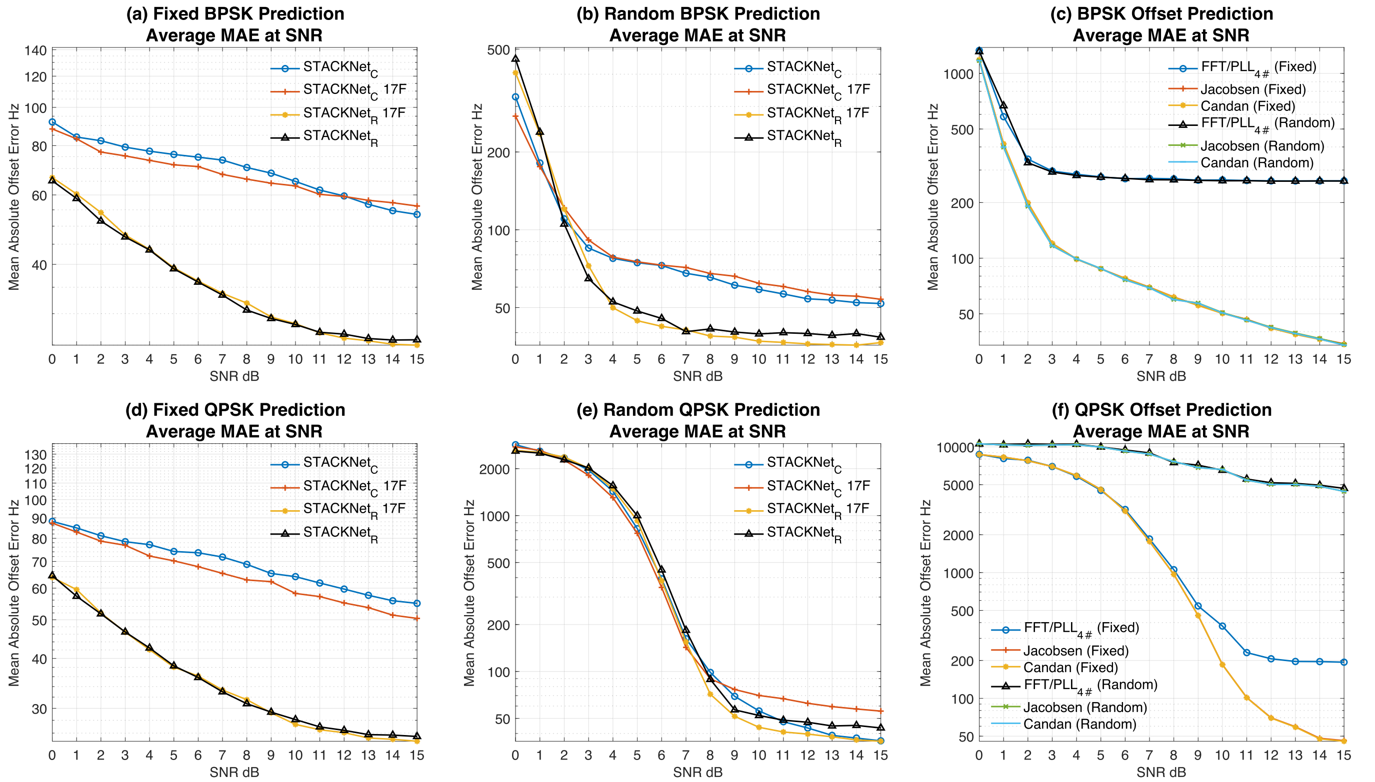


FIGURE 4. Comparison between stacked model configurations for classification (STACKNet$_C$) and regression (STACKNet$_R$) demonstrates better performance for CFO estimation on short BPSK and QPSK sequence lengths of 104 samples verse FFT/PLL methods.

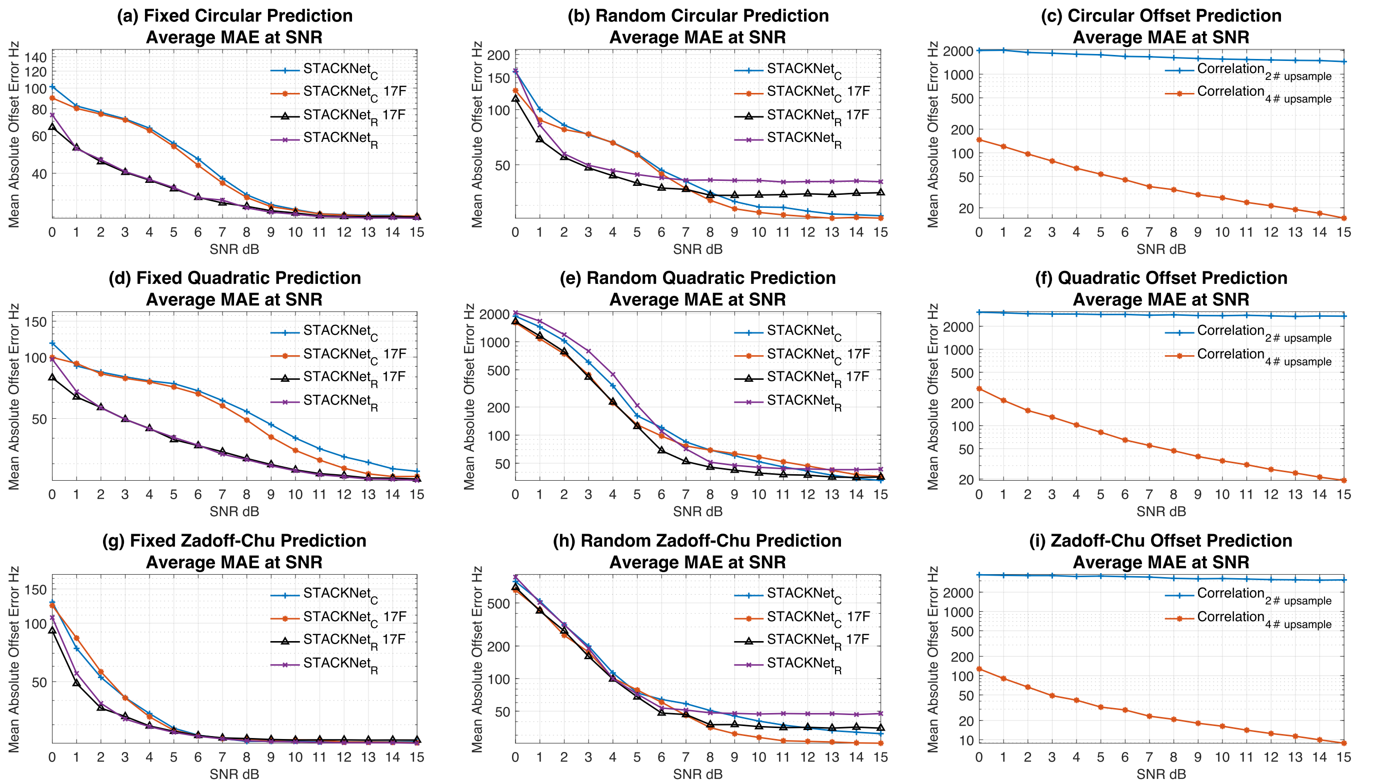


FIGURE 5. The stacked model demonstrates higher accuracy on the chaotic map than the cross-correlation with $2\times$ upsampling however does not perform as well as the $8\times$ upsampled cross-correlation.

Figure 7 displays a box plot for the execution timing of each method. The network is more complex than the conventional FFT/PLL, and this is reflected in the timings, hence the trade-off between accuracy and complexity. It is

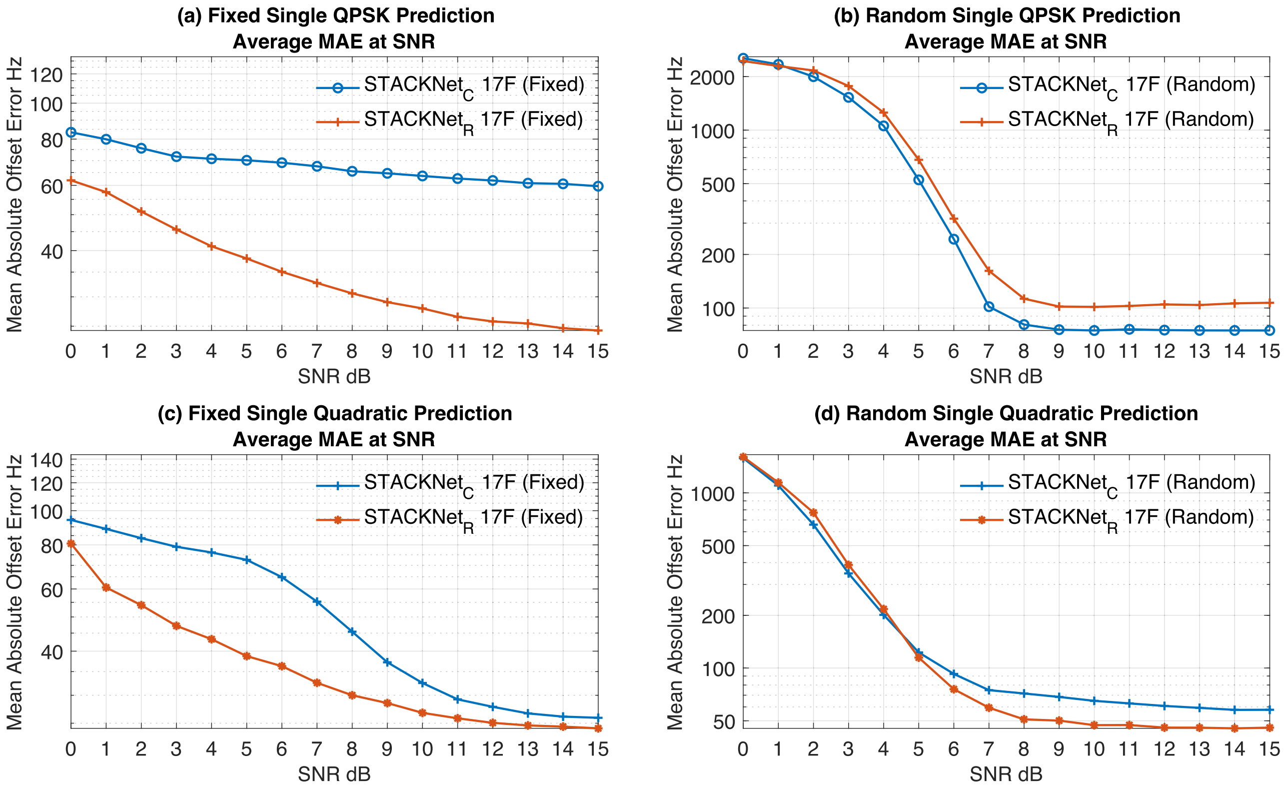


FIGURE 6. Models trained on a single modulation exhibit similar MAE to those trained on multiple modulations, indicating that training on multiple modulations does not appear to influence the performance of the model as much as the choice of modulation itself.

TABLE 5. Comparison between STACKNet$_C$ and STACKNet$_R$ on random sequences indicates an improvement in MAE Hz when data augmentation is applied and a small improvement in MAE Hz resulting from classification as opposed to regression.

Comparison	Mean Difference in MAE Hz
$\text{MAE}_{\text{STACKNet}_C 17F} - \text{MAE}_{\text{STACKNet}_C}$	25.7
$\text{MAE}_{\text{STACKNet}_R 17F} - \text{MAE}_{\text{STACKNet}_R}$	26.9
$\text{MAE}_{\text{STACKNet}_C} - \text{MAE}_{\text{STACKNet}_R}$	6.1
$\text{MAE}_{\text{STACKNet}_C 17F} - \text{MAE}_{\text{STACKNet}_R 17F}$	4.9

notable that it takes longer to process a single record on a DL model than it does to process a batch size of 100 records. This is due to the hardware environment being more suited to parallel execution, which will be an important consideration when integrating DL into other systems. Such an estimate may be taken as an average across windowed sequences for the received signal. The brute force cross-correlation method is much more expensive than the other two given the wide frequency range.

Those models constructed with data augmentation demonstrate an improvement over those learning from the unprocessed signal in the randomised setting. Both variants of the models (classification and regression) appear consistent in the influence of each of the features shown in Figure 8. One notable difference is that they disagree on the influence of the lagged difference for the conjugate of the signal where the imaginary value does not contribute as highly to the model

FIGURE 7. Execution speed of the simpler FFT/PLL method is faster in comparison to the deep network model which performs well on larger batches and is faster than the brute force cross-correlation method.

accuracy for the classification model STACKNet$_C 17F$ as opposed to the regression model. Variables contributing the lowest scores include the signal raised to 4th power and the lag-1 difference of phase in the signal. Both models nominate the phase of the squared signal $\angle r^2$ as causing the highest MAE when the feature is replaced with Gaussian noise. The low resolution FFT (length of 104), appears to be influential to both models, however is not able to be used in isolation

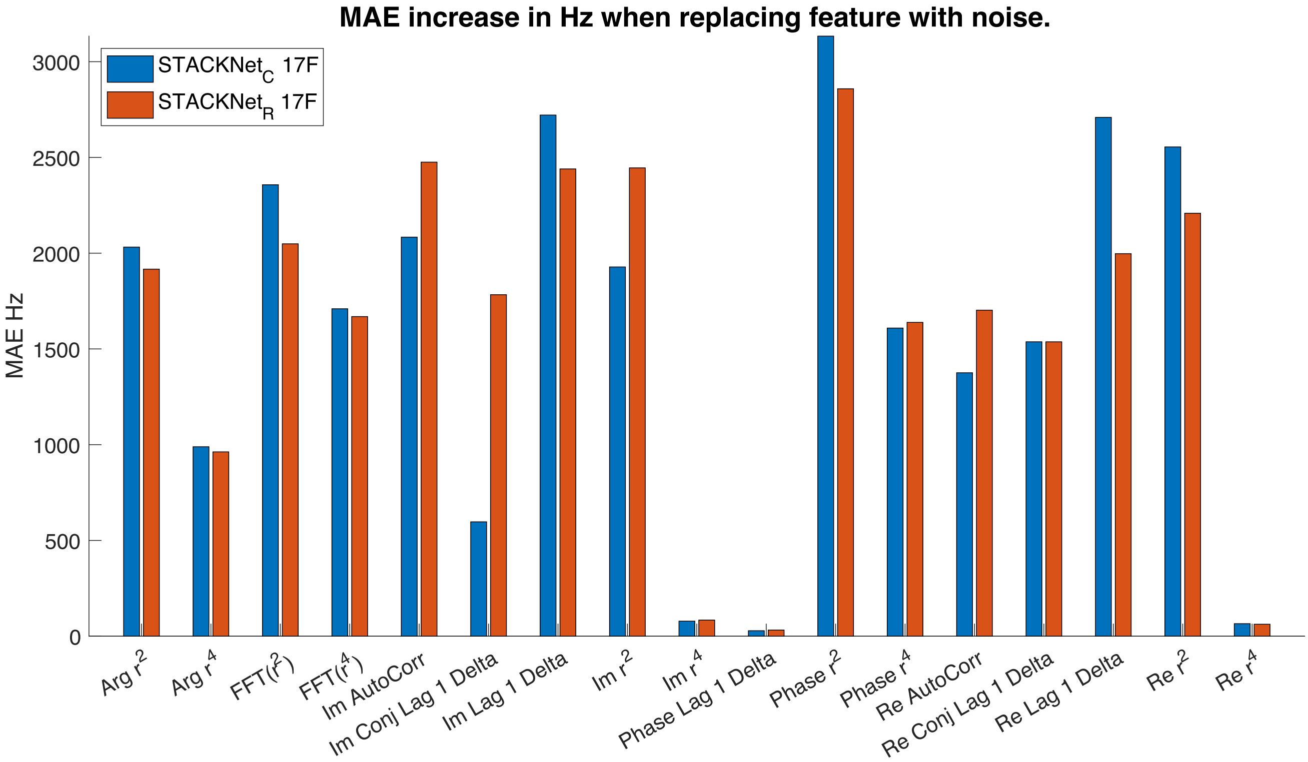


FIGURE 8. **Feature importance indicated by the increase in MAE Hz when the feature is replaced with gaussian noise.**

from the auto-correlation and squared polar form of the signal.

A. DISCUSSION

After training the proposed stacked network on the selected set of modulations, the model was able to produce more accurate CFO estimates than the FFT/PLL and the cross-correlation methods for short message sequences. On the other hand, the cross-correlation method required a longer message sequence to outperform the DL model. As shown in the related research, DL is capable of CFO estimation for short random sequences [2] and for noisy sinusoidal modulations [3], [18]. The stacked network models are also able to accept random sequences of several chaotic maps without reference to a template pilot sequence, indicating the ability of the trained network to estimate CFO without explicit knowledge of the feedback parameters for these types of signals. As such, this methodology is suitable for use with chaotic modulations and, given the ability to estimate frequency offset, it may be possible for such a method to estimate additional parameters required for chaotic synchronisation, such as the time dependent state variables of the chaotic map. Future research in this task may investigate the use of encoder-decoder networks in the estimation and tracking of multiple chaotic system parameters such as in [29].

Data augmentation was applied to the model, and in the randomised setting, demonstrated an improvement of approximately 20 Hz MAE over those models which did not make use of data augmentation. In the fixed preamble setting, data augmentation did not demonstrate much influence over the performance of the model, this is indicative that the variation in message content is influential over the performance of the model, with a fixed preamble illustrating low variation (outside of the channel model) as opposed to randomised sequences. While DL is capable of representation learning without the requirement of manual feature engineering, it is also true that domain specific feature engineering does provide an advantage in the application of DL. Such an approach indicates that DL will be most useful where it can be incorporated into communications systems alongside conventional signal processing methods in a hybridised form.

In this study we applied simulations with an AWGN channel to generate the required data. The difficulty in the supervised learning approach is the requirement for off-line training, which requires a large volume of data especially when training across multiple signal modulations. The amount of data required increases with each supported modulation so as to ensure an equal sized population for each modulation in the training set. However this research has not investigated the potential for transfer learning [16] to enable the network to adapt to new modulations or channel models, which is a topic for future investigation.

Performance of the model is influenced by the modulation of the signal as shown in the results, hence the network model is learning features related to the modulation in the carrier offset estimation task. In an end-to-end learning setting, it may be possible to dynamically learn a suitable modulation to reduce receiver error as demonstrated in works such as [30] and [31]. Future work will investigate methods of incorporating learnt CFO estimation which may jointly benefit from the modulations learnt at the transmitter, necessarily moving from an offline supervised learning problem to an online learning problem.

Execution timing demonstrated that the DL model is more efficient on batches of signal frames rather than on a single signal frame. This is also a result consistent with the benchmarking performed in [32]. This poses a design challenge for the practical application of DL models in communications systems, where batches of signal frames will be necessary to most efficiently make use of the DL architecture. Future work will be required to investigate the practical implementation challenges of integrating DL based CFO estimation within an end-to-end wireless communications system.

IV. CONCLUSION

In this article we have demonstrated the use of a stacked sequence-to-sequence encoder to perform carrier frequency offset estimation in multiple modulations, including for feedback dependent chaotic maps. The proposed architecture has been shown to outperform FFT/PLL and cross-correlation methods on short sequences, in both the fixed preamble setting and in the randomised setting without knowledge of the modulation, and in the randomised setting without a pilot template. However increasing the message sequence length did enable the cross-correlation method to outperform the DL model, at the expense of additional execution time. Data augmentation in the randomised setting, was shown to provide an increased accuracy for the CFO estimation (of approximately 20 Hz) and indicates that while DL models are capable of learning feature representations directly from raw IQ values, the use of appropriately chosen features is an avenue for enhancing the performance of the model. Iterative estimation was performed by separate stages of the stacked network architecture with an error correction performed at the final stack, thereby taking advantage of the composability of DL modules as a means of iteratively refining the CFO estimate. This work demonstrates the capability of DL techniques to estimate the carrier offset parameter for chaotic communications, and provides an incremental step towards the application of DL in short messaging systems and chaotic communication.

REFERENCES

[1] W. G. Cowley, "Phase and frequency estimation for PSK packets: Bounds and algorithms," *IEEE Trans. Commun.*, vol. 44, no. 1, pp. 26–28, Jan. 1996.

[2] R. M. Dreifuerst, R. W. Heath, Jr., M. N. Kulkarni, and J. Charlie, "Deep learning-based carrier frequency offset estimation with one-bit ADCs," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, May 2020, pp. 1–5.

[3] H. R. Almayyali and Z. M. Hussain, "Deep learning versus spectral techniques for frequency estimation of single tones: Reduced complexity for software-defined radio and IoT sensor communications," *Sensors*, vol. 21, no. 8, p. 2729, Apr. 2021.

[4] J. Zhang, T. Q. Duong, R. F. Woods, and A. J. Marshall, "Securing wireless communications of the Internet of Things from the physical layer, an overview," *Entropy*, vol. 19, no. 8, p. 420, 2017.

[5] G. Kolumbán, M. P. Kennedy, and L. O. Chua, "The role of synchronization in digital communications using chaos. II. Chaotic modulation and chaotic synchronization," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 45, no. 11, pp. 1129–1140, Nov. 1998.

[6] B. Jovic, *Synchronization Techniques for Chaotic Communication Systems*. Berlin, Germany: Springer, 2011.

[7] A. J. Michaels, "Digital chaotic communications," Ph.d. dissertation, Georgia Inst. Technol., Atlanta, GA, USA, 2009.

[8] J. Tao, L. Yang, and X. Han, "Enhanced carrier frequency offset estimation based on Zadoff–Chu sequences," *IEEE Commun. Lett.*, vol. 23, no. 10, pp. 1862–1865, Oct. 2019.

[9] D. C. Rife and R. R. Boorstyn, "Single tone parameter estimation from discrete-time observations," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 5, pp. 591–598, Sep. 1974.

[10] M. D. Macleod, "Fast nearly ML estimation of the parameters of real or complex single tones or resolved multiple tones," *IEEE Trans. Signal Process.*, vol. 46, no. 1, pp. 141–148, Jan. 1998.

[11] B. G. Quinn, "Estimating frequency by interpolation using Fourier coefficients," *IEEE Trans. Signal Process.*, vol. 42, no. 5, pp. 1264–1268, May 1994.

[12] B. G. Quinn, "Estimation of frequency, amplitude, and phase from the DFT of a time series," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 814–817, Mar. 1997.

[13] E. Jacobsen and P. Kootsookos, "Fast, accurate frequency estimators [DSP tips & Tricks]," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 123–125, May 2007.

[14] U. Orguner and C. Candan, "A fine-resolution frequency estimator using an arbitrary number of DFT coefficients," *Signal Process.*, vol. 105, no. 10, pp. 17–21, Dec. 2014.

[15] A. Host-Madsen and P. Handel, "Effects of sampling and quantization on single-tone frequency estimation," *IEEE Trans. Signal Process.*, vol. 48, no. 3, pp. 650–662, Mar. 2000.

[16] Y. Bengio, Y. LeCun, and G. E. Hinton, "Deep learning for AI," *Commun. ACM*, vol. 64, pp. 58–65, Jun. 2021.

[17] G. Izacard, B. Bernstein, and C. Fernandez-Granda, "A learning-based framework for line-spectra super-resolution," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3632–3636.

[18] V. Ninkovic, A. Valka, D. Dumic, and D. Vukobratovic, "Deep learning-based packet detection and carrier frequency offset estimation in IEEE 802.11ah," *IEEE Access*, vol. 9, pp. 99853–99865, 2021.

[19] Mathworks. *Commun.Coarse Frequency Compensator Compensate for Frequency Offset of PAM, PSK or QAM Signal*. Accessed: Sep. 29, 2022. [Online]. Available: https://au.mathworks.com/help/comm/ref/Commun.coarsefrequencycompensator-system-object.html

[20] Y. Wang, K. Shi, and E. Serpedin, "Non-data-aided feedforward carrier frequency offset estimators for QAM constellations: A nonlinear least-squares approach," *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 13, Dec. 2004, Art. no. 856139.

[21] Mathworks. *Commun.Carriersynchronizer Compensate for Carrier Frequency Offset*. Accessed: Sep. 29, 2022. [Online]. Available: https://au.mathworks.com/help/comm/ref/Commun.carriersynchronizer-system-object.html

[22] M. Rice, *Digital Communications: A Discrete-Time Approach*. Upper Saddle River, NJ, USA: Prentice-Hall, 2009.

[23] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[26] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 550–558.

[27] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 464–472.

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

[29] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 45, pp. 22445–22451, Nov. 2019.

[30] T. J. O'Shea, K. Karra, and T. C. Clancy, "Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2016, pp. 223–228.

[31] F. A. Aoudia and J. Hoydis, "End-to-End learning of communications systems without a channel model," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 298–303.

[32] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64270–64277, 2018.

CHRISTOPHER P. DAVEY (Graduate Student Member, IEEE) received the master's degree in information technology from the Queensland University of Technology (QUT), Australia, in 2007, and the M.Sc. degree in mathematics and statistics from the University of Southern Queensland (USQ), Australia, in 2020, where he is currently pursuing the Ph.D. degree in deep learning for wireless communications. He has over a decade of professional experience in software development and systems integration. He has worked on "Artificial Intelligence for Decision-Making (AI4DM)" and "AI-Enabled Communicating Systems" research project funded by the Australian Government's Department of Defence.

ISMAIL SHAKEEL (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electronic engineering from the University of South Australia, in 1997, the two master' degrees from the University of Canterbury (NZ) and Monash University, in 2001 and 2002, respectively, and the Ph.D. degree in telecommunications, in 2007. He joined the Defence Science and Technology Group (DSTG), in 2011. He is currently with the Information Sciences Division, DSTG, and also an Adjunct Professor with the University of Southern Queensland. Before joining DSTG, he has worked in both academia and industry and holds a patent and generated more than 40 technical reports and publications in the field of telecommunications. His current research interests include signal detection and classification techniques, artificial intelligence-enabled wireless communication, interference-resistant signalling, chaotic communication, and cooperative wireless communication.

RAVINESH C. DEO (Senior Member, IEEE) leads Advanced Data Analytics Research Laboratory as a Professor at the University of Southern Queensland (USQ), Australia. He is a Clarivate Highly Cited Researcher with publications ranking in top 1% by citations for field and publication year in the Web of Science citation index and is among scientists and social scientists who have demonstrated significant broad influence, reflected in the publication of multiple papers frequently cited by their peers. He leads cross-disciplinary research in deep learning and artificial intelligence having supervised more than 30 Ph.D./M.Sc. degrees. He has published more than 270 articles, 150 journals, and seven books with a cumulative citation that exceeds 11,100. He received the Employee Excellence Awards, the Elsevier Highly Cited Paper Awards, and the Publication Excellence and Teaching Commendations.

SANCHO SALCEDO-SANZ was born in Madrid, Spain, in 1974. He received the B.S. degree in physics from the Universidad Complutense de Madrid, Madrid, in 1998, the Ph.D. degree in telecommunications engineering from the Universidad Carlos III de Madrid, Madrid, in 2002, and the Ph.D. degree in physics from the Universidad Complutense de Madrid, in 2019. He spent one year at the School of Computer Science, University of Birmingham, U.K, as a Postdoctoral Research Fellow. He is currently a Full Professor with the Department of Signal Processing and Communications, Universidad de Alcalá, Spain. He has coauthored more than 220 international journal articles in the field of machine learning and soft-computing and its applications. His current research interests include soft-computing techniques, hybrid algorithms, and neural networks in different problems of science and technology.

JEFFREY SOAR is currently a Personal Chair in human-centered technology with the School of Business, University of Southern Queensland. He came to academic research from a long and distinguished career in industry including as the Chief Information Officer in government agencies in Australia and New Zealand. His research interests include AI, e-business, e-health, technology and development, and social and organizational change.

4.3 Links and Implications

This publication demonstrated the ability for the same DL model to estimate CFO for multiple modulations. It also demonstrated evidence that the estimation accuracy is influenced by the modulation type, with DL model being most accurate on BPSK, Circular chaotic map and Zadoff-Chu map while both QPSK and the Quadratic chaotic map produced larger MAE. It was shown that the DL was able to produce more accurate results on fixed sequences as opposed to random sequences. However, the network also outperformed conventional methods on both fixed and random sequences. This article addresses a gap in the literature by providing evidence for DL being able generalise estimation over multiple modulations, and randomised sequences. At the same time, it demonstrates a novel branching DL architecture that is able to refine its parameter estimate in two stages. The article also investigates features derived from the conventional CFO estimation processes to show that the DL method benefits by incorporating processing stages from conventional methods and produces significant improvements over those methods.

CHAPTER 5: PAPER 2 - END-TO-END LEARNING OF ADAPTIVE CODED MODULATION SCHEMES FOR RESILIENT WIRELESS COMMUNICATIONS

5.1 Introduction

This chapter presents a copy of the article published in the Elsevier journal Applied Soft Computing (vol 159 (2024), p 111672, ISSN 1568-4946, doi:10.1016/j.asoc.2024.111672. `https://doi.org/10.1016/j.asoc.2024.111672`).

E2E learning jointly optimises transmitter and receiver based on the AE architecture. However, this approach is only able to learn a single code rate for the resulting coded modulation. AMC is an approach to adaptation which can select from multiple coded modulation schemes in response to channel conditions. This publication proposes a novel E2E approach to learning AMC by modifying the DL architecture based on multi-task learning. It also provides a customised training algorithm and demonstrates the ability of the model to produce gains over several conventional codes in AWGN and Rayleigh fading channels. The modification of the AE architecture to support multiple codes and the accompanying novel training algorithm eliminate the need to train several independent AE for each code rate and enables E2E learning for AMC schemes.

Research Highlights

- Proposes an end-to-end machine learning architecture for generating coded modulation schemes with different data rates.

- Custom training/multi-task learning produces coded modulation schemes with competitive error-rate performance.

- Proposed approach outperforms several traditional coding techniques for short codes.

- Proposed approach is versatile in adapting to Rayleigh fading channel condition without model retraining.
- The proposed end-to-end machine learning architectures have practical benefits for developing resilient wireless communication systems.

5.2 Published Article 2

Applied Soft Computing Journal 159 (2024) 111672

Contents lists available at ScienceDirect

Applied Soft Computing

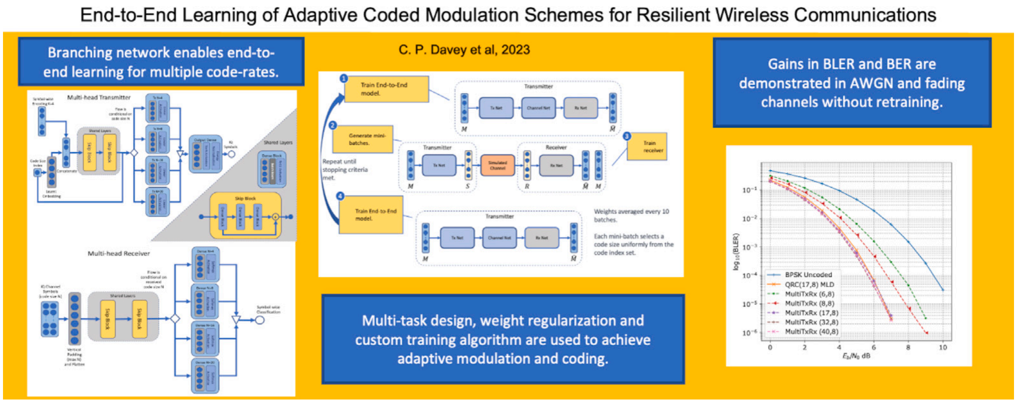
journal homepage: www.elsevier.com/locate/asoc

End-to-end learning of adaptive coded modulation schemes for resilient wireless communications

Christopher P. Davey [a,*], Ismail Shakeel [a,b], Ravinesh C. Deo [a,*], Ekta Sharma [a], Sancho Salcedo-Sanz [c], Jeffrey Soar [d]

[a] *School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, 4300, Queensland, Australia*
[b] *Spectrum Warfare Branch, Information Sciences Division, Defence Science and Technology Group (DSTG), Edinburgh, 5111, SA, Australia*
[c] *Department of Signal Processing and Communications, Universidad de Alcalá, Alcalá de Henares, 28805, Spain*
[d] *School of Business, University of Southern Queensland, Springfield, 4300, Queensland, Australia*

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Wireless communications
Adaptation
Coding design
Deep learning
Multi-task learning

ABSTRACT

Adaptive modulation and coding schemes play a crucial role in ensuring robust data transfer in wireless communications, especially when faced with changes or interference in the transmission channel. These schemes involve the use of variable coding rates, which can be achieved normally through code puncturing or shortening, and have been adopted in 4G and 5G communication standards. In recent works, auto-encoders for wireless communications have demonstrated the ability to learn short code representations that achieve gains over conventional codes. Such a methodology is attractive as it can learn optimal representations under a variety of channel conditions. However, due to its structure the auto-encoder does not currently support multiple code rates with a single model. This article draws upon the discipline of multi-task learning, as it applies to deep learning and therefore devises a branching architecture for the auto-encoder and custom training algorithm in training transmitter and receiver for adaptive modulation and coding. In this article we aim to demonstrate improvements in Block Error Rate over conventional methods in the Additive White Gaussian Noise channel, and to analyse the performance of the model under Rayleigh fading channels without retraining the auto-encoder on the new channel. This article demonstrates a novel approach towards training

* Corresponding authors.
E-mail addresses: christopher.davey@usq.edu.au (C.P. Davey), ravinesh.deo@usq.edu.au (R.C. Deo).

https://doi.org/10.1016/j.asoc.2024.111672
Received 4 April 2023; Received in revised form 14 November 2023; Accepted 15 April 2024
Available online 26 April 2024

auto-encoder models to jointly learn adaptive modulation and coding schemes framed as a multi-task learning problem. The research outcomes extend end-to-end learning approaches to the design of adaptive wireless communications systems.

1. Introduction

Adaptive modulation and coding (AMC) methods can enable wireless communication systems to optimise the transmission of data over a channel with varying operating conditions, message sizes, and data transfer rates. This involves adjusting the modulation scheme and error correction coding rate in real-time based on the channel conditions. AMC algorithms are designed to monitor the channel conditions in real-time and dynamically select a suitable modulation and coding scheme that provides the best trade-off between data rate and error protection for the given channel conditions. The selection of the best modulation and coding scheme is normally made by a look-up table or an algorithm that maps the channel conditions to a particular modulation and coding scheme with respect to transmit power, expected channel use and error rate [1]. The modulator/demodulator and encoder/decoder algorithms for each modulation and code are normally designed and implemented separately on the communication platform [2]. This paper focuses on using machine learning techniques to generate multiple coded modulation schemes from a unified model architecture for channel adaptation.

In communications systems, the primary goal is to transmit a message through a communication channel to a receiver and then reconstruct the original message without error at the receiver. Distortions that are introduced by the channel are the primary obstacle to an error free recovery of the transmitted message.

Fig. 1 illustrates a simple wireless communications system, which is the focus of our article. In such a system, a message M, defined in bits, is formatted and communicated by a transmitter over the air (the channel) to be recovered and interpreted at the receiver. In order to be transmitted, messages may be coded for error correction and modulated for transmission over the channel. The modulation process converts a bit sequence into a waveform where each discrete point (symbol) in the waveform is represented as a series of complex symbols $x(t) \in \mathbb{C}$, having both in-phase and quadrature (IQ) components, where t indicates the discrete time step for the symbol. We use the term "symbol" after modulation because a single symbol can refer to more than 1 bit of the original message. The number of bits mapped to a symbol is referred to as the order of the modulation. However, a modulation is not necessarily sufficient to allow the receiver to recover the message without error.

The subject of this research is coding, which is an essential component of any communication system that enables error detection and correction at the receiver end. A variety of techniques are available to code a particular message, and these include linear block codes and convolutional codes. The resulting code words are longer than the original message block, and the ratio between the original message at length K and the resulting N bit code word is known as the code rate K/N. For smaller code rates, one requires more symbols to be transmitted across the channel. The ability to perform error correction using a coded message can achieve significant improvements over uncoded messages (a coding gain) but this comes with a trade-off in terms of the number of usages required to transmit the message, or in other words, the amount of power required to send the message. However, the ability to receive a message without error reduces the need for re-transmissions.

As illustrated in Fig. 1, the coding and modulation are an integral part of the system. Therefore in this article, we focus on the learning of the coding and modulation process that assumes a perfect synchronisation of the transmitted signal with the receiver. The coding and modulation stages in the wireless communications system are typically designed in isolation of each other, and often, they do not account for the distortions introduced by different types of channels. This method of system design is referred to as the block design, where components are individually optimised and do not consider interactions between components or the channel distortion [3].

The ability to automatically learn each of the stages within a wireless communications system presents an advantage over the block design approach, since each of the stages can be optimised jointly with respect to a given channel condition and hardware imperfection. As such, learning how to transmit and receive coded information over the wireless channel has recently attracted significant attention in the field of wireless communications. Deep learning (DL) methods, and in particular the AE architecture have been demonstrated to jointly optimise both transmitter and receiver with respect to an assumed channel model [2]. Such a joint approach, learns coding and modulation in an end-to-end manner by gradual optimisation of the model parameters to minimise the error produced in symbol-wise classification of individual messages [2].

In end-to-end learning, a transmitter acts as an encoder network to encode a message, while the receiver acts as the decoder network to retrieve the original message [2]. The channel is represented either as an instantaneous function which adds perturbations to the output of the transmitter [2], may be learnt through adversarial techniques [4] or reinforcement learning (RL) is applied without assuming a channel [5]. The design of the AE for wireless communications in [2], does not support learning more than one code rate, in that approach support for multiple code rates requires separate networks. Therefore the subject of this research article investigates how to parameterise and alter the structure of a single AE for wireless communications to enable support for multiple code rates.

Multi-task learning (MTL) in DL research is concerned with the challenge of training a single network architecture to concurrently perform different but related tasks, which may also have a dependency relation between them [6]. Approaches to MTL consider the architecture design, model regularisation and training methods [6]. There is a relationship between negative transfer in MTL and catastrophic forgetting which occurs in sequential learning [7,8]. Negative transfer occurs when certain tasks negatively impact the ability to learn other tasks when learnt concurrently [6]. Hence regularisation techniques acting to minimise negative transfer during training are a key concern of MTL. This is realised through the design of the network architecture (hard sharing) as well as through weight regularisation and loss functions (soft sharing) [6]. In this article we approach AMC from the perspective of MTL and apply both hard sharing in the AE network design as well as soft sharing in the approach to regularisation during the training procedure. We demonstrate that a multi-branching variant of the AE is better suited to learning AMC in comparison to a single path AE network. The model is trained by iterating between end-to-end and receiver only training, and we apply a weight averaging regularisation technique [9] to improve the error rates for each of the resulting code rates.

The main contributions of this paper are:

- To change the structure of the AE for wireless communications to enable learning multiple code rates with a single neural network architecture. A shared path with branching output heads are activated based on a selected code rate parameter to support end-to-end learning for AMC.
- To frame the end-to-end learning of AMC for wireless communications as a multi-task learning problem.
- Propose a training procedure which iterates between end-to-end training and receiver training, producing lower error rates than single step training.

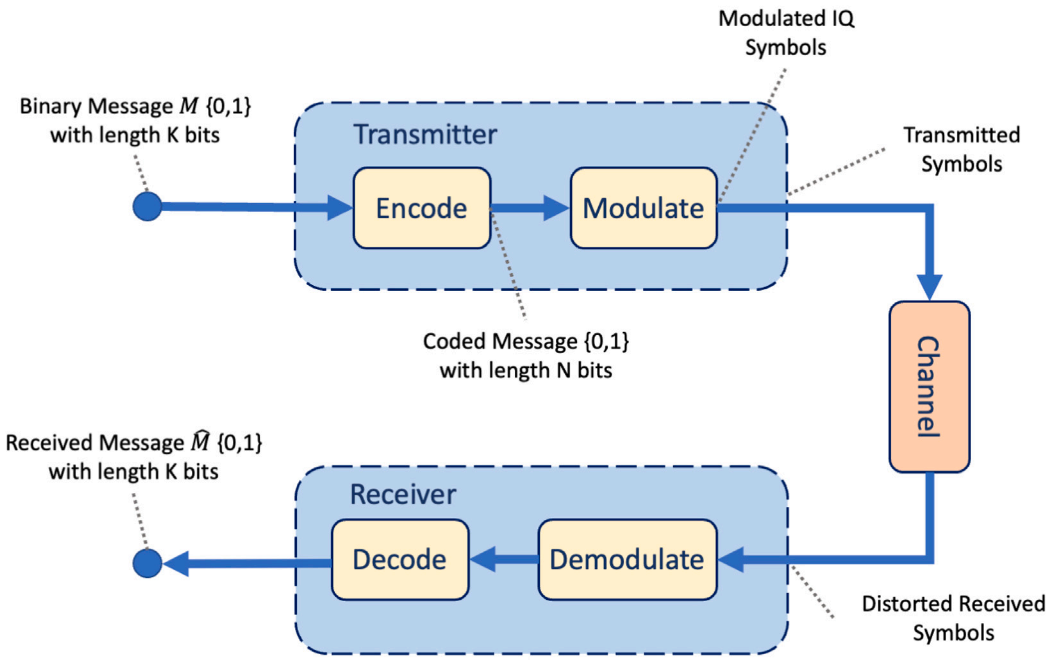


Fig. 1. A simplified schematic of a typical wireless communication system. In the transmitter, a binary message of K information bits is coded as N bits for error correction during the Encode stage. The Modulation stage converts the bits into discrete complex symbols using amplitude, phase or frequency to differentiate bits. On the receiver side, the demodulator converts the received modulation symbols to the original code word of length N and the decoding block converts from the code back to the original K information bits of the message.

- To show that the proposed method achieves results closely matching and improving upon performance of maximum likelihood decoding (MLD) with conventional codes over several code rates and channels.

The remainder of this article is structured in the following manner: Section 2 provides an overview of the research into end-to-end learning for wireless communications as well as describing multi-task learning and adaptive modulation and coding. Section 3 describes the multi-rate AE, the method of regularisation and its custom training procedure. Section 4 reports on the BER and BLER of the model under several channel environments. Section 5 describes the limitations of the model and discusses the generalisation capability. The article concludes in Section 6 where the summary of findings is given and further directions for research are proposed.

2. Background and related work

The use of AE neural networks for learning an end-to-end wireless communications system was first proposed in [2]. An AE was demonstrated to learn optimal short codes for the AWGN channel. The AE model architecture as described by the article is illustrated in Fig. 2. The model consists of symbol-wise encoding for 2^K possible messages, a transmitter containing multiple dense blocks (layers) and an energy normalisation constraint, transmitter output IQ symbols for a code size N, an assumed channel function, and a receiver (also containing multiple dense blocks) whose task is to predict the received message index [2]. In a typical AE the middle layers are applied to find a compressed set of features for the input, whereas in the wireless communications design, the middle layers typically represent the output of the transmitter, which are influenced by the distortion provided by the channel function. By applying DL to the design of wireless communications systems the article introduced the potential use of generalised hardware platforms such as graphical processing units (GPUs), and enabled the opportunity for optimisation against complex channels without requiring an analytic mathematical model for the channel [2]. The article demonstrated that an AE with a code rate of $K/N = 4/7$ could achieve equal performance to MLD decoding for the Hamming(7,4) code and that an AE trained on an interference channel could achieve lower error rate than a quadrature amplitude modulation (QAM) time sharing modulation scheme for equivalent code sizes [2]. However, the design of the classifier architecture is limited to a fixed number of message bits K and a fixed code size N, and is constrained to the domain of short length burst transmissions. Such codes are beneficial for use in energy constrained communications such as in the internet of things (IoT). The AE model as presented in [2] serves as the canonical model for DL end-to-end communications systems under simplified constraints. Related work stemming from this initial research extends the AE architecture and examines applications to end-to-end learning, over-the-air learning, and the use of custom training algorithms.

A key assumption of the AE model as described in [2] requires that a differentiable channel function for a given channel is predefined to permit back-propagation between the transmitter and receiver. An alternate approach is to train a generative adversarial network (GAN) using observed perturbations from the true channel, thereby removing the assumption of a predetermined channel. The approach described in [4], applies this technique to approximate an unknown channel function and provides a fully differentiable channel model for training of the AE. The architecture is trained and evaluated on several channels including the AWGN, Rayleigh Fading and Frequency Selective multipath channels. Under the AWGN and Rayleigh fading channels, the AE-GAN model is compared with the end-to-end AE from [2], as well as conventional coding methods, demonstrating the effectiveness of the GAN in approximating the channel during training [4]. Each component (transmitter, channel GAN and receiver) is trained in succession using an iterative algorithm, where components not participating in each training cycle had their weights frozen [4]. The architecture assumed a constant size of K message bits for the AE input and output, and while it was able to approximate bit-wise output leveraging Convolutional Neural Network (CNN) layers to support differing message lengths, it did not address the effect of altering the code size N and was not designed to produce multiple code rates without retraining.

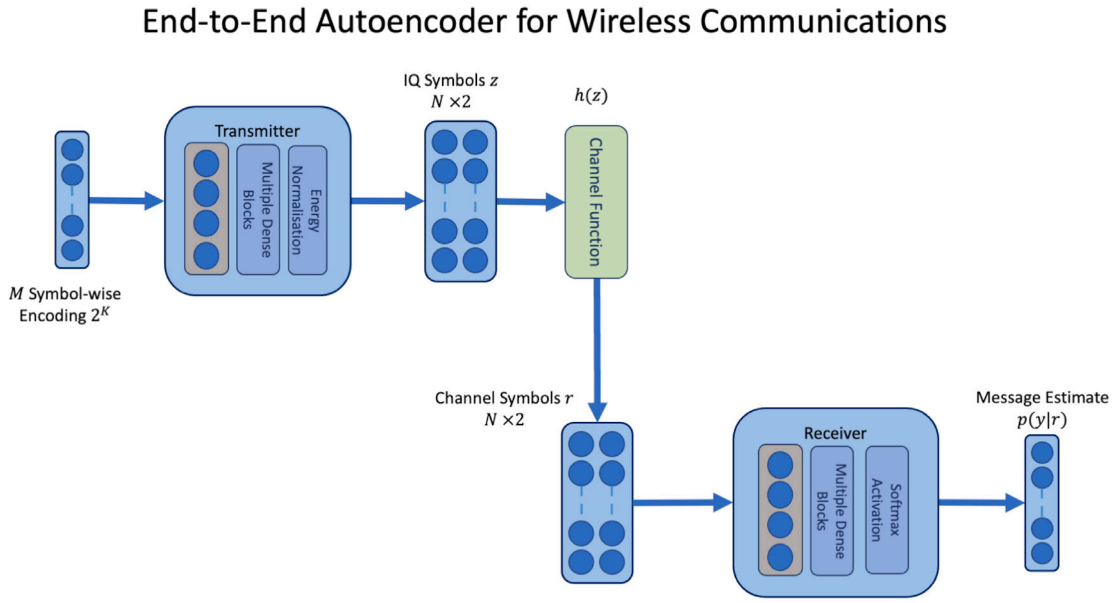


Fig. 2. The neural network architecture of the end-to-end AE for wireless communications described in [2]. The architecture of transmitter and receiver each consist of multiple dense layers or blocks, and the transmitter includes an energy normalisation layer. The model is defined for a predefined number of message bits K and a single code rate N, as well as an assumed channel function $h(z)$. The transmitter learns to encode message M from one of 2^K messages into IQ symbols which are sent over the assumed channel function and the receiver learns to estimate the probability of the message M given the received output of the channel r.

Another method for addressing the channel assumption was proposed by leveraging an iterative training algorithm based on RL in [5, 10]. The training algorithm first trains the receiver by back-propagation and in the second step applies the receiver loss to an approximation of the gradient to perform back-propagation at the transmitter [5]. The advantage of this method is that it is agnostic to the true channel, although does require reliable feedback of losses from the receiver, and applied an equalisation method from [2] in order to train against the Rayleigh Block Fading (RBF) channel [5]. This latter point indicates the dependency of DL methods on the perturbations that are applied to their training data, in wireless communications systems, these perturbations result from the channel environment. In [10] the RBF channel is evaluated with and without equalisation, indicating a slight difference in performance between the approaches. Changes made to the channel, outside of that applied during training, will have a negative impact on the performance of the model, hence further investigation of approaches to regularisation for end-to-end training are required for adaptability to changing channel conditions. While the research focus for AE in wireless communications has addressed the assumed channel function, we propose that adaptability can be achieved by also considering an AMC scheme.

One approach to address adaptation, post training, is to deploy and retrain the receiver independently of the transmitter on the true channel, also referred to as tuning or transfer learning. This technique is described in [3] where an iterative approach is applied to train the AE end-to-end and secondly to fine-tune the receiver. The method is demonstrated on a software defined radio (SDR) implementation during an over the air (OTA) training phase [3]. OTA training allows a more realistic channel environment as opposed to end-to-end training, and requires additional stages to support synchronisation such as filtering, timing, phase, and carrier frequency offset corrections [3]. Two separate sub-networks were incorporated prior to the receiver/decoder model to correct for timing and phase offsets as well as learning new features to assist in decoding [3]. While the trained system did not perform as well as the comparative communications system, the work demonstrates that receiver tuning OTA has the potential to improve the adaptation of the overall system post end-to-end training and emphasises the mismatch between the analytic and actual channel models. This is further evidence of the sensitivity of DL models to perturbations during training. While we do not investigate an OTA implementation, we do investigate the performance of an end-to-end AE on several channels without retraining or tuning, and propose that an architecture capable of generating AMC schemes is advantageous in environments for which it was not initially trained.

Extensions of the AE architecture have been made to incorporate concatenated coding techniques for reliable communications in [11] where the AE learns the inner code and the outer code is implemented with a low density parity check (LDPC) code. Such an outer code is capable of variable code rates, independent of the AE model. In this method, the AE performs bit-wise encoding and decoding as opposed to the classification of a symbol-wise output in [2]. The role of the receiver is to estimate the log-likelihood ratio for each bit and is trained in a manner equivalent to maximising the achievable rate of the communications system [11]. Both the encoder and decoder are parameterised by the signal to noise ratio (SNR) and it is shown that learnt constellations are correlated with the given SNR [11]. The association of constellation and SNR enables a form of adaptive coding which does not vary the code length, but may rearrange the constellation points instead. In our approach we instead modify the AE model architecture to allow parameterisation for a code index which permits the model to learn a mapping between the code index parameter and a variable code length, thereby achieving AMC by varying code rates.

Given an assumed channel, and a measure of the communication error rate, it is possible to iteratively search for an optimal code rate. A technique for this type of search is presented in [12]. The main contribution of the article is first to address the issue of overfitting in the end-to-end AE and propose an additional regularisation term that maximises the mutual information between the transmitter symbols and the output of an assumed channel function [12]. This regularisation term is applied to the loss term of the AE and is approximated by training a separate neural network. The search algorithm, described as capacity driven AE, iterates over multiple SNR and trains AEs at incremental code rates K/N until improvement in the mutual information over previous AE falls below a given threshold [12]. However, long training durations, and the limitations around sampling for large message bit lengths are a disadvantage for AE. An exhaustive search is feasible for short messages, but less feasible as K increases. The ability to design a single network architecture that can support multiple code rates could reduce the overall duration of such a search algorithm. To make these changes to the AE architecture, we propose framing the task of training multiple code rates as a MTL problem.

MTL seeks to regularise a network to perform several related but distinct tasks through the network architecture (hard sharing) or through regularisation methods constraining weights in matching layers (soft sharing) [6]. The simplest hard sharing approach uses a common single path with multiple outputs to demonstrate that the relatedness between tasks benefits network regularisation through the transfer of inductive bias between those tasks [13]. This type of architecture also has the advantage of limiting the number of parameters required by multiple networks, since a common path is shared between separate branches rather than requiring an individual network for each task [13]. Training of such architectures is performed while learning multiple tasks simultaneously, whereas in our approach we train successively for single tasks (code rates) using a common architecture. However successive training on different tasks is well known to suffer from catastrophic forgetting [7] also termed negative transfer in the MTL literature [6]. The challenge of MTL for sequential learning is approached in [14] which proposes a dynamic architecture comprising of shared and task specific paths which is trained in a sequential manner. This approach is demonstrated to reduce the negative transfer between tasks on the shared path [14]. During training and inference the structure of the network is altered dynamically and enables the execution of one task at a time [14], in this manner the parameterisation of each task is implicit to the current organisation of the network. In our approach we define tasks as code rates and dynamically reconfigure the network structure during training and inference while supplying a code index parameter to indicate which code rate the transmitter should output. To regularise the shared path between tasks we make use of a simple weight averaging regularisation [9].

Adaptation for both modulation and coding has been demonstrated to achieve more reliable communications under varying levels of interference when compared to adaptation for modulation only [15]. AMC-enabled systems have also been shown to produce higher data transfer rates over various communication environments [16,17]. AMC is implemented by selecting a combination of modulation scheme and error-correcting code to achieve a target BER under a given SNR partition [16,18]. Different code sizes may be constructed from the same family of codes so that the minimum distance of the code remains constant over varying SNR and channel fading conditions [19]. Such codes can be formed by shortening, that is reducing both information and code word bits, or puncturing by removing some of the parity check bits from each code word [20]. Cyclic codes are well suited to shortening and puncturing since the original decoding procedure can be applied to the resulting code [20]. This category of codes includes the Hamming code [21], Bose–Chaudhuri–Hocquenghem (BCH) code [21,22] and the quadratic residue code (QRC) [21]. Rather than shortening a family of codes, we augment the end-to-end AE for wireless communications to jointly learn multiple code rates. The advantage of jointly learning modulation and coding would enable AMC schemes to be tuned specifically for target channel conditions.

3. Methodology

In our work we consider the AE architecture in [2] as the canonical DL architecture for jointly learning modulation and coding in a wireless communications system. However the structure of the canonical AE is limited to a single message bit size K and a single code size N. In this article we make several alterations to the original AE architecture to support end-to-end learning for AMC with multiple code sizes N. We modify the network architecture so that it is able to learn several predefined code sizes by adding branching outputs at the transmitter and receiver. We also add a parameter to select the code size index during training and inference in the transmitter. In the main path of the network we include skip connections [23] between blocks of dense units to enable a slightly larger network to aid in learning multiple code sizes.

This section includes the details for the changes to the AE architecture (Section 3.1), the approach to training for the transmitter and receiver models (Section 3.2) and the selected channel functions that are applied during training and evaluation (Section 3.3).

3.1. Model architecture

The AE, described in [2], consists of a single path through the network and a channel function implemented as an AWGN layer (Fig. 2). Estimation is performed as a classification for the corresponding one-hot encoded input message M. One-hot encoding represents each binary message M by defining an input vector of the same length as the number of unique binary messages (in 2^K messages). Each unique message is represented as a 1 at its corresponding index in the input vector (a value from 0 to $2^K - 1$) with all other positions of the vector set to 0. Symbol-wise classification is performed at the receiver by learning to estimate the probability of a given message at the corresponding vector index $p(M|r(t))$ given a set of channel symbols (in-phase and quadrature values) $r(t)$. The index with the highest probability is mapped from the index to the estimated binary message $\hat{M} = \arg\max p(M|r(t))$.

The branching structure of our proposed model is inspired by hard sharing in MTL. In the simplest form of hard sharing, a common network path is followed by a set of branches that correspond to each distinct task [6]. The common path learns shared features for all tasks. In our architecture a task, corresponding to a network branch, represents a different code rate K/N due to the change in the code size N. The common path contains two dense blocks composed of a feed-forward layer, batch-normalisation [24] and either rectified linear unit (ReLU) [25] or Swish [26] activation functions. The two dense blocks feature residual connections to form skip blocks which assist in preventing extremes in the gradient during back-propagation of deeper networks [23]. Although these networks are relatively shallow, we have found that the skip connections do improve the performance of the network.

In the transmitter, the common path accepts a concatenation between the one-hot encoded message and an embedding representing the code index. The code size index is provided to a discrete gate that determines which branch of the network architecture should receive features from the common path during the forward pass. Each branch contains a feed-forward layer followed by a linear activation. The output of the transmitter consists of a feed-forward layer followed by a tanh activation and an energy normalisation layer that is applied prior to the channel function. An overview of the transmitter model is shown in Fig. 3. The architecture is parameterised by a set of code rates $i \in \{1_1, \dots, i_n\}$ that are used by the branch node to select which of the output branches are active during training and inference. The branch node is indicated by the *Discrete Gate* in the transmitter, Fig. 3 and the receiver, Fig. 4.

The architecture of the receiver is illustrated in Fig. 4 and follows a similar pattern to that of the transmitter. Instead of receiving an additional code size parameter during inference, the length of the received channel symbols is stored at the input to the network. Zero padding is applied to the received symbols up to the maximum allowed code size. The padding layer is followed by a series of skip blocks having the same structure as those in the transmitter, prior to the discrete gate. The *Discrete Gate* receives the stored length parameter and uses this to determine which output branch (*Dense Layer i*) to activate on the forward pass. Each output branch consists of a feed-forward layer and a soft-max activation layer.

The number of units in the input layers of the transmitter relate to the 2^K possible messages while for the receiver the input units depend on the code size that is selected in the transmitter. To determine the number of units for the shared path of each network, a stepwise approach was applied. Starting from the value of K the number of units

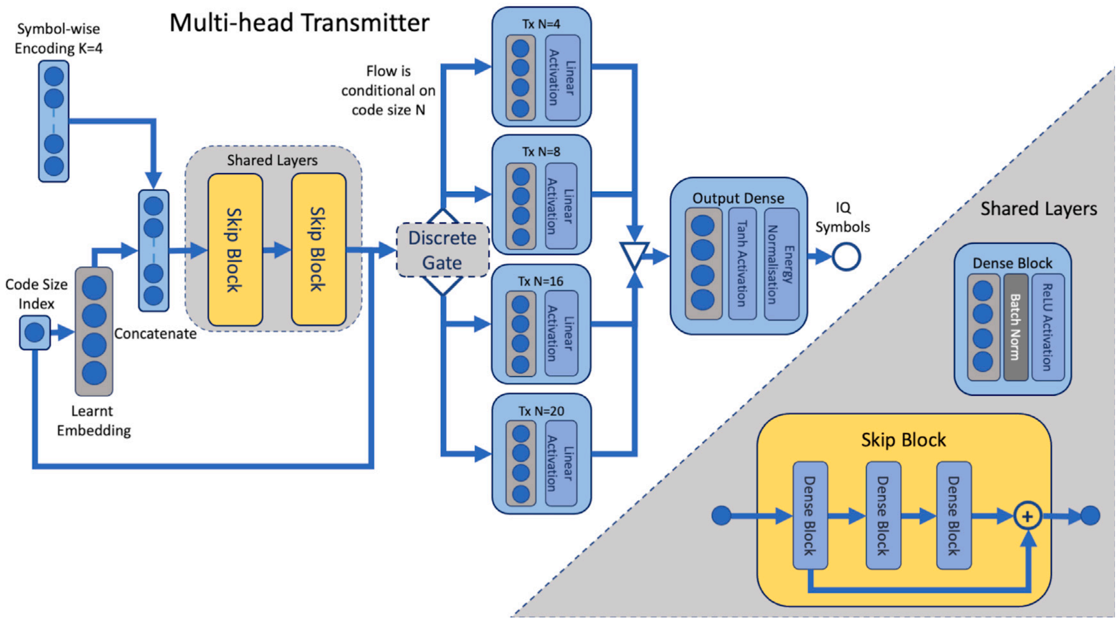


Fig. 3. The transmitter contains a common path followed by a discrete gate which switches between the set of selected code sizes prior to merging and energy normalisation. The transmitter sizes for N provide an example of how each branch represents a different code rate, and are an example of 4 bit model configuration.

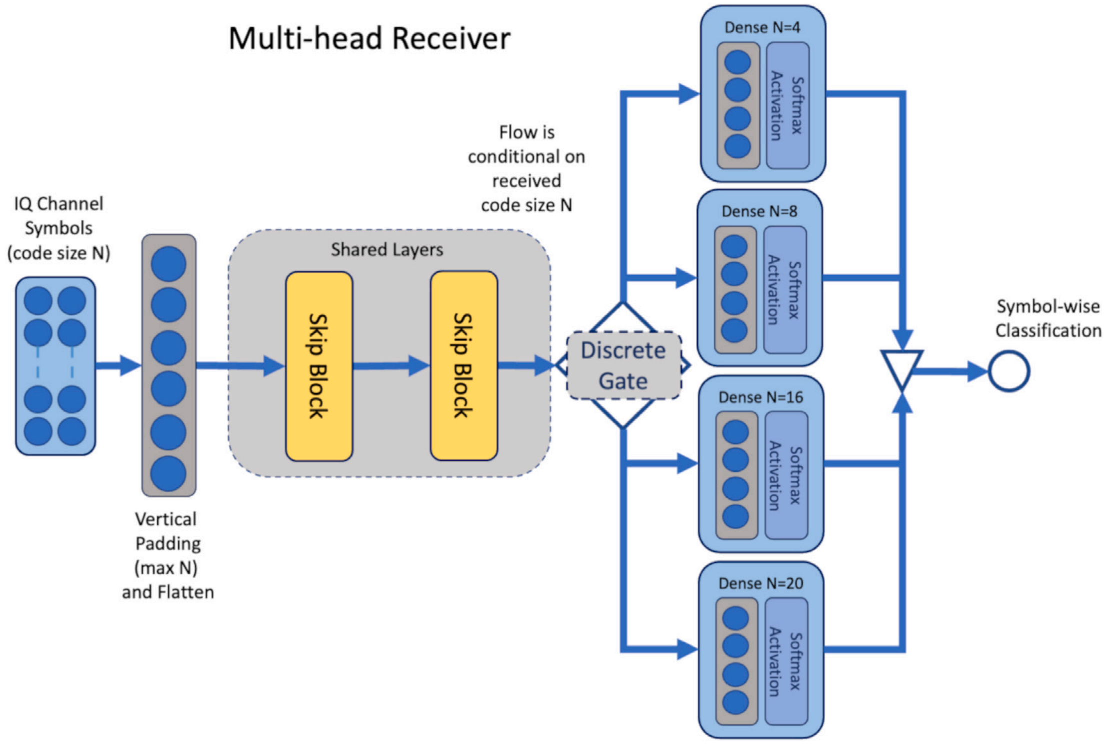


Fig. 4. The receiver architecture follows a similar approach to the transmitter where the shared path is followed by a discrete gate and a separate classification layer corresponding to each code size.

was gradually increased until an acceptable performance was achieved during training. The list of layers and the number of units in each layer of the transmitter network is shown in Table 1 and is shown for the receiver network in Table 2.

Each branch in the transmitter and receiver represents a code of size N. The code size parameter was supplied to the network as a choice from a set of code sizes $i \in \{i_i, \dots, i_n\}$. Several variants of the network architecture were trained to evaluate the effect of additional code sizes (or increased number of tasks) on the overall performance of the model. The branching transmitter and receiver networks were trained for three message sizes, $K = 4$ bits, $K = 7$ bits and $K = 8$ bits. For each case, the network was trained to map K information bits to multiple code sizes of N channel symbols for transmission. For the $K = 4$ bit message size, the code sizes included $N = 4, 8, 16$ and 20, similarly for the $K = 7$ bit message size, code sizes $N = 11, 15,$ and 34 and finally for the $K = 8$ bit message configuration, code size $N = 6, 8, 17, 32$ and 40 were selected. The configurations for each model, message and code size are listed in Table 3. The table also lists the total number of trainable parameters in each neural network.

Table 1
The number of units in each layer of the transmitter model. The list of choices for code size indices i are provided as a parameter to the architecture. During training and inference the one-hot encoded message and the selected code size index i are provided as input to the network.

Layer	Units $K = 4$	Units $K = 7$	Units $K = 8$	Group
Input 1 $M = 2^K$ units	16	128	256	Input layers
Input 2 code size index i	1	1	1	
Code index embedding $2K$ units	8	14	16	
Concatenate $2K + 2^K$ units	24	142	272	
Dense layer	32	512	512	Skip block
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Dense layer	64	64	64	
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Dense layer	32	512	512	
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Dense layer	32	512	512	Skip block
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Dense layer	64	64	64	
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Dense layer	32	512	512	
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Gate layer	–	–	–	
Dense layer	$i = 4$, units = 8	$i = 11$, units = 22	$i = 6$, units = 12	Code size $i \in \{i, \dots, i_n\}$ branches
Dense layer	$i = 8$, units = 16	$i = 15$, units = 30	$i = 8$, units = 16	
Dense layer	$i = 16$, units = 32	$i = 34$, units = 68	$i = 17$, units = 34	
Dense layer	$i = 20$, units = 40	–	$i = 32$, units = 64	
Dense layer	–	–	$i = 40$, units = 80	
Linear activation	–	–	–	Transmitter output for code size i
Reshape layer	$[2 \times i]$	$[2 \times i]$	$[2 \times i]$	
Dense layer	$[2 \times i]$	$[2 \times i]$	$[2 \times i]$	
Tanh activation	–	–	–	
Energy normalisation	–	–	–	

We compare the different model configurations against several conventional codes. The $K = 4$ bit model is compared with the MLD performance of a system that uses uncoded binary phase shift keying (BPSK) modulation with extended Hamming (8,4) code. The 7 bit model is compared with three BCH coded systems, two of which use shortened codes s-BCH(11,7) and s-BCH(34,7) derived from mother codes BCH(15,11) and BCH(63,36) respectively, with the additional code being BCH(15,7). The K = 8 bit model is compared with uncoded BPSK and a QRC(17,8) code.

In both architectures, the gate function at layer l, $f_g^{(l)}$ is parameterised by the code rate index i and input to the current layer $h^{(l)} = f^{(l-1)}\left(h^{(l-1)}\right)$ which selects from a set of branches comprising the next layer $f_i^{(l+1)} \in \left\{f_0^{(l+1)}, f_1^{(l+1)}, \dots f_n^{(l+1)}\right\}$ (Eq. (1)). In the transmitter, the code rate index is supplied as an explicit parameter to the network, whereas in the receiver the code rate index is determined based on the number of symbols received from the channel. During the forward pass only one path through the branch is active (at the branch layer, the active branch will be $f_i^{(l+1)}$). During back-propagation, no gradients exist for the inactive branches, hence only the active branch receives the gradient update. Each of the respective shared paths in the transmitter and receiver, participate in back-propagation.

$$f_g^{(l)}\left(i, h^{(l)}\right) = f_i^{(l+1)}\left(h^{(l)}\right),\quad \text{where } f_i^{(l+1)} \in \left\{f_0^{(l+1)}, f_1^{(l+1)}, \dots f_n^{(l+1)}\right\} \tag{1}$$

3.2. Training algorithm

It is important to consider a suitable regularisation approach to prevent negative impact on overall network performance between tasks. This is achieved with two approaches, randomised sampling for code size during training, and weight regularisation. Training consists of mini-batches (32 messages per batch) and code sizes are selected from a random uniform distribution each mini-batch. The update of each model is performed with back-propagation each mini-batch and the gradient is calculated for the selected code size. Over the course of learning, the weights for each layer in the network are stored and are averaged across mini-batches every ten iterations. This latter approach to regularisation is based on the stochastic weight averaging (SWA) performed in [9], and in the results we have observed that training using SWA produces better performance as opposed to those networks trained without SWA. SWA combined with a cyclical learning rate schedule [27] is demonstrated in [9] to improve the generalisation of the network. During training back-propagation is performed with the Adam optimiser [28] combined with the cyclical learning rate between 0.0001 and 0.001.

In addition to the sampling scheme and SWA regularisation, an alternating training algorithm is applied in four steps described in Algorithm 1 and Fig. 5. These steps consist of: (1) train the end-to-end network, (2) generate mini-batches using the transmitter, (3) train the receiver against a simulated channel and record the loss, and (4) update the end-to-end network. In Step 1, back-propagation is run on the end-to-end model which contains both the receiver and transmitter models. This updates the weights in both the receiver and transmitter models, and the AWGN channel is simulated directly as part of the end-to-end model architecture. During Step 2, the transmitter is used to generate the transmitter symbols and the channel is simulated independently of both transmitter and receiver models. In Step 3, the receiver is then trained using the channel response as the receiver input, and during back-propagation the receiver loss is calculated against the true messages. This allows the transmitter and receiver to be evaluated independently and the resulting receiver loss is used to coordinate

Algorithm 1 During training, the end-to-end model is trained iteratively with the receiver model.

```
Input: epochs                         ▷ The number of training iterations.
Input: batchSize                      ▷ The size of each training or validation batch.
Input: transmitModel                  ▷ The transmitter model.
Input: receiveModel                   ▷ The receiver model.
Input: endToEndModel                  ▷ The end-to-end model containing transmitter, channel and receiver models.
Input: channel                        ▷ The channel simulation function.
Input: snr                            ▷ An initial SNR dB for perturbation of training data.
Input: codeSizeList                   ▷ The set of allowed code lengths.
Output: endToEndModel                 ▷ The end-to-end model updated after training.

  loss ← ∞
  weightsList ← []

  for i ← 1, epochs do
      if Train with random SNR then
          snr ← Random-Uniform(0,9)                 ▷ Use randomised SNR to perturb data for training.
      end if
      codeSize <- Random-Uniform(codeSizeList)      ▷ Randomly select the code size for the training batch.
      Train-EndToEnd(batchSize, codeSize, snr, endToEndModel)                   ▷ Fig. 5 (Step 1)
      messages ← Transmit-Samples(batchSize, codeSize, snr, transmitModel, channel)  ▷ Fig. 5 (Step 2)
      receiverLoss ← Train-Receiver(messages, codeSize, receiveModel)          ▷ Fig. 5 (Step 3)
      if receiverLoss < loss then
          loss ← receiverLoss
          Save(endToEndModel, transmitModel, receiveModel)  ▷ Save models each time learning improves at the receiver.
      end if
      if i mod 10 equals 0 then
          w ← Average-Weights(weightsList)
          Set-Weights(endToEndModel, w)             ▷ Apply weight averaging every 10 iterations.
      else
          w ← Get-Weights(endToEndModel)
          Append(weightsList, w)
      end if
      Train-EndToEnd(batchSize, codeSize, snr, endToEndModel)                   ▷ Fig. 5 (Step 4)
  end for

  return endToEndModel
```

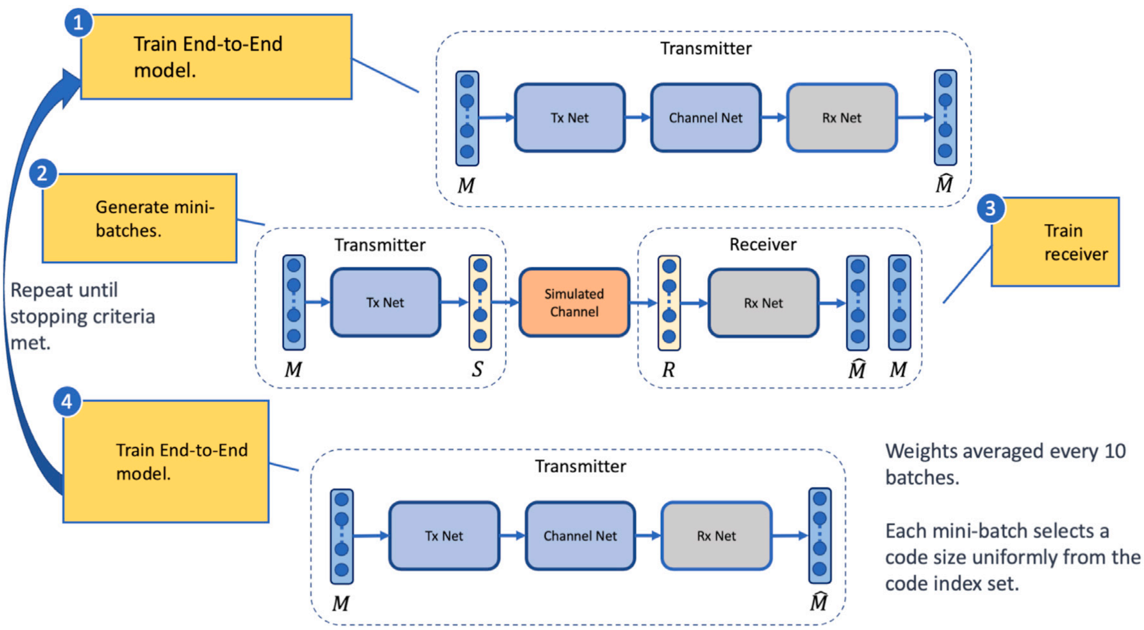


Fig. 5. Custom training algorithm consisting of several stages interleaving training of receiver and end-to-end model.

Table 2
The number of units in each layer of the receiver model. The list of choices for code size indices i are provided as a parameter to the architecture, and selected at runtime based on the length of the received channel symbols.

Layer	Units $K = 4$	Units $K = 7$	Units $K = 8$	Group
Input 1 $[2 \times i]$ units	$[2 \times i]$	$[2 \times i]$	$[2 \times i]$	Input from channel for code size i
Flatten Layer	$2i$	$2i$	$2i$	
Dense layer	32	512	512	Skip block
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Dense layer	64	64	64	
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Dense layer	32	512	512	
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Dense layer	32	512	512	Skip block
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Dense layer	64	64	64	
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Dense layer	32	512	512	
Batch normalisation	–	–	–	
Activation (ReLU or Swish)	–	–	–	
Gate layer	–	–	–	
Dense layer 2^K units	16	128	256	Output $i \in \{i, \dots, i_n\}$ Branches
Softmax activation	–	–	–	

Table 3
Multiple variations of the model are trained with separate configurations for message bits K and code size N. The total trainable parameters for each neural network counts all weights, biases, and batch normalisation parameters. The final column lists the conventional codes included in comparisons of BER and BLER selected channel conditions.

Configuration	Model variant	K	N	Transmitter parameters	Receiver parameters	Comparison codes
1	4 bit single model	4	$i \in \{4\}$	3831	4880	Uncoded BPSK
2	4 bit multi-rate model	4	$i \in \{4, 8, 16, 20\}$	14 471	13 888	Uncoded BPSK extended Hamming(8,4)
3	7 bit multi-rate model	7	$i \in \{11, 15, 34\}$	671 151	767 616	s-BCH(11,7) BCH(15,7) s-BCH(34,7)
4	8 bit multi-rate code	8	$i \in \{6, 8, 17, 32, 40\}$	649 125	1 101 696	Uncoded BPSK QRC(17,8)

intermittent checkpointing of both models. Finally, Step 4 updates the end-to-end network after weight averaging before the next training iteration.

3.3. Channel functions

The assumed channel function that is applied during training of the proposed model is the AWGN channel. Evaluation for the BER and BLER is made on three channels, the AWGN and two variants of Rayleigh fading differing in duration, the first applies fading to the entire block (Block fading), and the second varies symbol to symbol (Bit fading). When evaluation is carried out, the proposed models are not retrained or tuned for the two additional fading channels.

In AWGN (Eq. (2)) additive Gaussian noise $n(t)$ is added to the output of the transmitter $z(t)$, where t is the discrete time step of the transmitter output.

$$r(t) = z(t) + n(t) \tag{2}$$

Eq. (3) shows the Rayleigh fading coefficient $a(t)$, at each discrete time t, applied to the transmitted signal $z(t)$, prior to addition of additive noise $n(t)$. The fading coefficient $a(t) = \frac{1}{\sqrt{2}}|a|e^{j\psi}$, is drawn from a complex standard normal distribution $a \sim \mathcal{CN}(0, 1)$, and it's argument multiplied with the exponential waveform with phase parameter ψ, we assume a constant phase $\psi = 0$. The duration of the coefficient varies under block or bit fading. In addition we assume no channel estimation to reverse the effect of fading on the receiver.

$$r(t) = a(t)z(t) + n(t) \tag{3}$$

The additive Gaussian noise $n(t)$ is drawn from the complex normal distribution $n(t) \sim \mathcal{CN}(0, \sigma^2)$. The variance σ^2 is derived from the desired SNR and the final output of the transmitter layer $z(t)$, having $t = [1 \cdots T]$ discrete time steps. A desired level of noise is first supplied to the channel simulation as the ratio of energy per bit to noise E_b/N_0 dB. To account for the selected code rate k/n, the E_b/N_0 dB is converted to the ratio of energy per symbol to noise E_s/N_0 dB $= E_b/N_0$ dB $+ 10log_{10}(k/n)$. The components for E_s and N_0 are then estimated from the transmitter symbols $z(t)$ where $E_s = \frac{\sum_{t=1}^{T} z(t)^2}{T}$ and $N_0 = \frac{E_s}{E_s/N_0}$. The parameter E_s/N_0 is also commonly referred to as SNR. The variance is then estimated as $\sigma^2 = N_0/2$ and used to sample from the complex normal distribution.

The output at the transmitter is normalised by the energy constraint $\|x\|_2^2 \leq 1$ implemented in Eq. (4) where $x(t) \in \mathbb{C}$ is the sequence of complex symbols output by the tanh activation layer and T the number of time steps in the sequence. During training it is possible to vary the SNR dB randomly or to train at a constant SNR dB. A fixed SNR of 6 dB performed best for the 4 bit message, however 7 bit and 8 bit messages

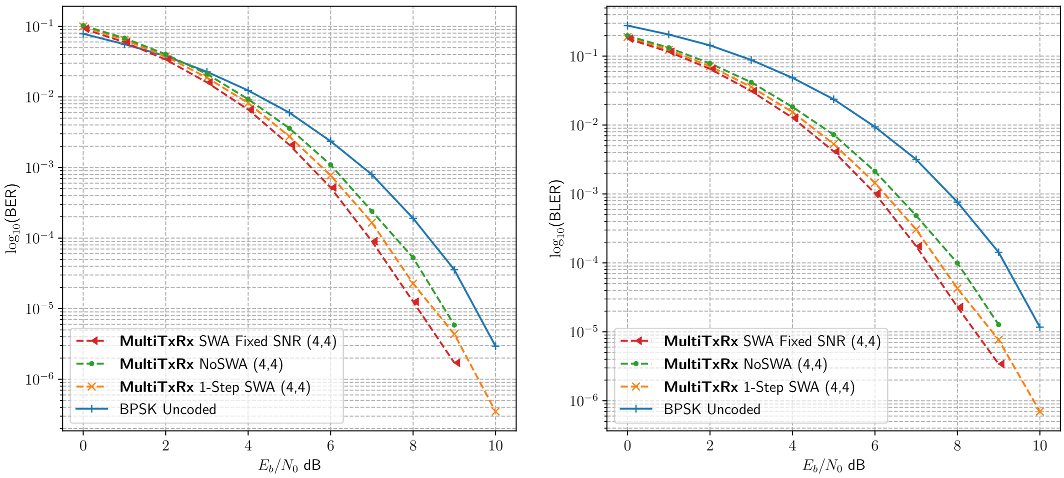


Fig. 6. BER and BLER in AWGN of several training methods, training with a standard back-propagation algorithm and SWA weight averaging (MultiTxRx 1-Step SWA), multi-step training without SWA (MultiTxRx NoSWA) and multi-step training with SWA and fixed SNR (MultiTxRx SWA Fixed SNR). Improvement in performance is indicated with the addition of SWA as well as when training with the multi-step training procedure as opposed to the standard back-propagation.

were trained with random SNR between 0 – 9 dB.

$$z(t) = \frac{x(t)}{\sqrt{\sum_{i=1}^{T} x(i)^2/T}} \tag{4}$$

4. Results

In this section we report the empirical evaluation for the proposed model at different code rates, listed in Table 3, for the AWGN and the two fading channels. In each case we refer to the proposed model as the *MultiTxRx* model to indicate a multi-branching transmitter and receiver. The first evaluation investigates the performance of the proposed training algorithm. The second set of evaluations reports on the performance of different variations of the architecture. These two sets of evaluations are used to examine the design choices for the model structure and training approach. After this we compare the set of code size configurations from Table 3 with the conventional codes in the AWGN channel, and evaluate performance without retraining or tuning in the fading channels.

The performance of several training algorithms are evaluated in Fig. 6, with message size $K = 4$ bits and code size $N = 4$ (from configuration 1, Table 3). Uncoded BPSK performance is included for reference. The *MultiTxRx 1-Step SWA* model was trained using standard back-propagation and included weight averaging. *MultiTxRx NoSWA* (no weight averaging) and *MultiTxRx SWA* where trained with the multi-step algorithm described in Fig. 5. The multi-step algorithm with weight averaging (*MultiTxRx SWA*) produces lower BER and lower BLER in comparison to standard back-propagation (*MultiTxRx 1-Step SWA*). Training without weight averaging produces higher BER and BLER.

Next, we compare the changes made to the AE architecture in Fig. 7 by training on multiple code sizes from configuration 2 in Table 3. The different types of architecture shown in Fig. 7 include a *Single Path* AE, *Single Tx MultiRx*, *MultiTx SingleRx*, *MultiTxRx* and *MultiTxRx Residual*. The *Single Path* model consists of a single common path in the network. The *Single Tx MultiRx* model applies a single path with a pooling layer to realise multiple codes in the transmitter, and classifies multiple codes using branching in the receiver. The structure is reversed in the *MultiTx SingleRx*. When trained with multiple code sizes, *MultiTxRx* is similar to the proposed architecture but does not feature skip connections and the *MultiTxRx residual* is the proposed architecture, including skip connections. The *MultiTxRx Residual* model performs better than the other models and is close to the extended Hamming(8,4) BER. Both versions of the *MultiTxRx* model exhibit similar BLER.

All code rates from configuration 2 in Table 3 are compared under in the AWGN channel in Figs. 8 and 9 and in the Block Fading and Bit Fading channels in Figs. 10 and 11 respectively. In the AWGN channel, it is difficult to see the difference in performance between code rates in relation to E_b/N_0 (except for $K = 4, N = 4$). In contrast, Fig. 9 displays the BER and BLER related to the energy per symbol (E_s/N_0) SNR dB. This example demonstrates that the smaller code rates can achieve lower BER and BLER as the channel noise increases at the cost of increased channel usage due to the increase in code size N. The aim of an AMC scheme is to maintain performance by trading off channel use in varying SNR.

In the Block Fading channel (Fig. 10) we observe that the BER is much higher than the uncoded BPSK while the BLER is much lower. The BER is higher because symbol-wise classification does not perform error correction of individual bits. An error on a code word may contain more incorrect bits in a single forward pass estimation. However, this approach achieves better BLER, as it can accurately classify, or map, the entire code word for a corresponding message. In the Block Fading channel, the entire code word is impacted by the channel fading. Bit-interleaving techniques can be applied in this circumstance which can produce an effect that is similar to a Bit Fading channel prior to decoding. The difference between the code rates is most noticeable in the Bit Fading channel (Fig. 11), performance improves as the code rate decreases (at the expense of channel use). In this channel, the $K = 4, N = 8$ code is slightly better than the baseline extended Hamming(8,4) code, as opposed to the AWGN channel. In the AWGN channel, code rate $K = 4, N = 8$ is close to the baseline extended Hamming(8,4) code, but differs slightly in higher SNR. The model is not retrained on either of the fading channels and is able to perform close to or better than the baseline.

Fig. 12 displays the performance of the $K = 7$ bit message and code rates from configuration 3 of Table 3 in the AWGN channel. The figure shows slight gains for BLER over the shortened s-BCH(11,7) and BCH(15,7) codes, and similar performance to the shortened s-BCH(34,7) code. There is less difference in BER performance for these codes. Under the Block Fading channel in Fig. 13 the BER is again higher, but the BLER is lower in comparison to the reference codes. In the Bit Fading channel, shown in Fig. 14, incremental gains are achieved on all code rates in comparison to the BCH and shortened codes (Fig. 14).

Configuration 4 from Table 3 for the $K = 8$ bit message and selected code sizes, is compared with the uncoded BPSK and the QRC code in the AWGN (Fig. 15), Block Fading (Fig. 16) and Bit Fading (Fig. 17) channels. In AWGN the BER produced by the model at the lower code

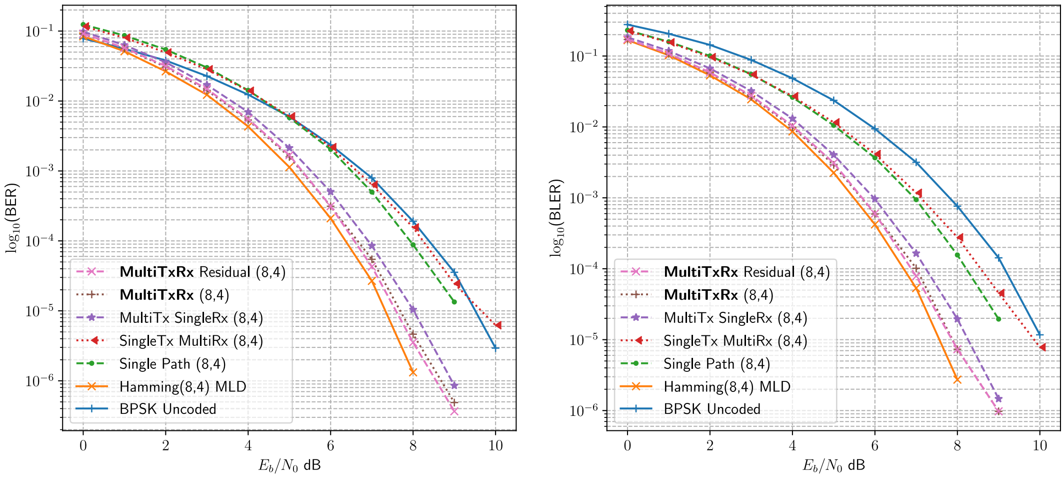


Fig. 7. The multi-branching Tx Rx architecture is compared with four variants of the architecture, a non-branching single path architecture (Single Path), a single branch transmitter with multi branch receiver (SingleTx MultiRx), the multi-branch transmitter and single receiver (MultiTx SingleRx) and multiple branching transmitter receiver with and without residual connections (MultiTxRx Residual and MultiTxRx). The choice of network architecture influences performance for the multi-task estimation of multiple code rates.

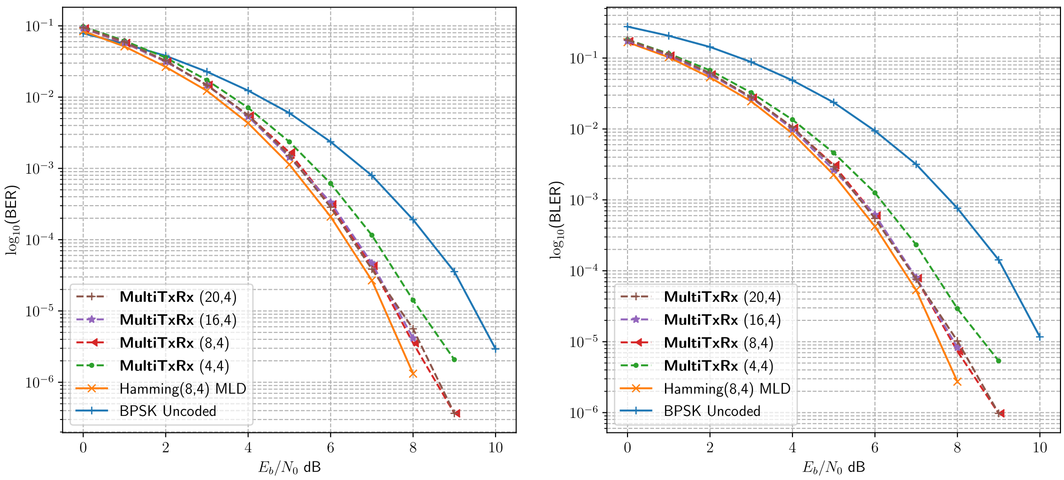


Fig. 8. BER and BLER in AWGN for MultiTxRx model with K = 4 bits and N = [4, 8, 16, 20] compared with BPSK uncoded and extended Hamming(8,4) maximum likelihood decoding (MLD).

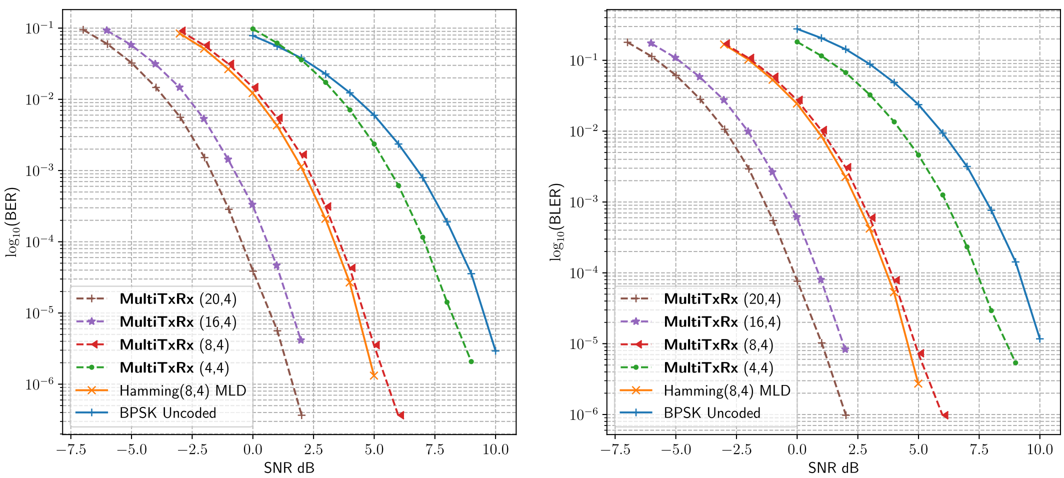


Fig. 9. The coding gain for each respective code rate is visible when plotting the BER and BLER in AWGN over the energy per symbol (E_s/N_0) SNR dB. The advantage of learning multiple codes enables operation under increased noise in the channel.

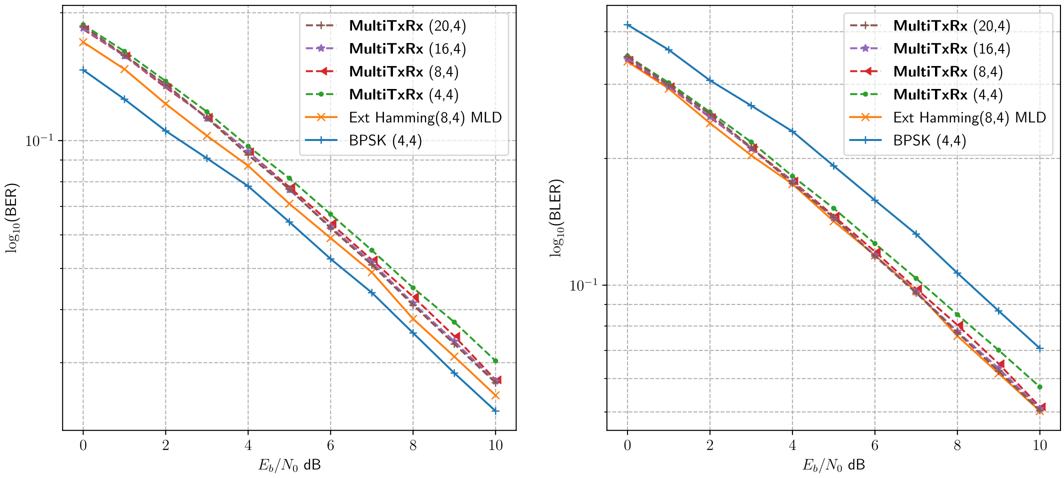


Fig. 10. BER and BLER in the Block Fading channel for MultiTxRx model with K = 4 bits and N = [4, 8, 16, 20] compared with BPSK uncoded and extended Hamming(8,4) maximum likelihood decoding (MLD). The proposed module was originally trained on the AWGN channel and is not trained for the Block Fading channel.

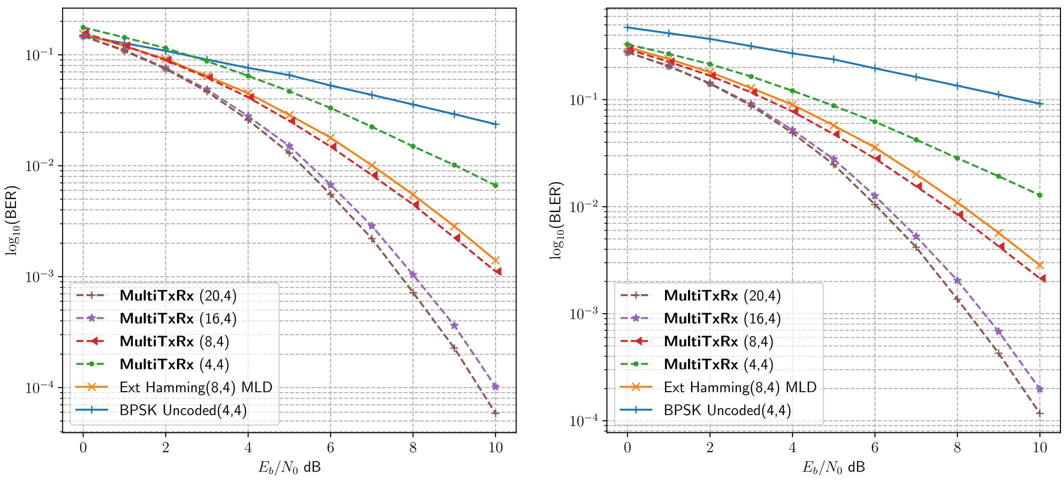


Fig. 11. BER and BLER in the bit fading channel for MultiTxRx model with K = 4 bits and N = [4, 8, 16, 20] compared with BPSK uncoded and extended Hamming(8,4) maximum likelihood decoding (MLD). The proposed module was originally trained on the AWGN channel and is not trained for the bit fading channel.

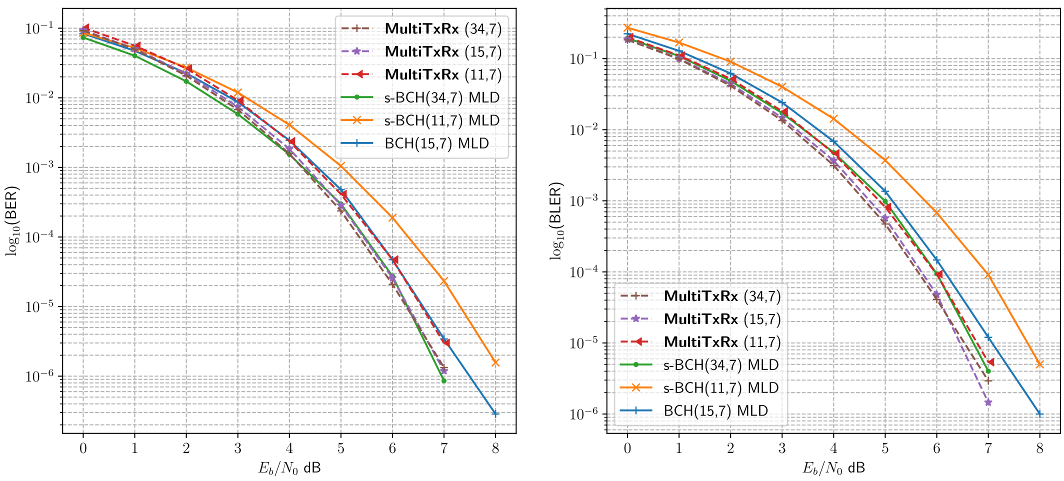


Fig. 12. BER and BLER in the AWGN channel for MultiTxRx model with K = 7 bits and N = [11, 15, 34] compared with shortened BCH codes s-BCH(11,7), s-BCH(34,7) and BCH code (15,7) maximum likelihood decoding (MLD), and trained with random SNR.

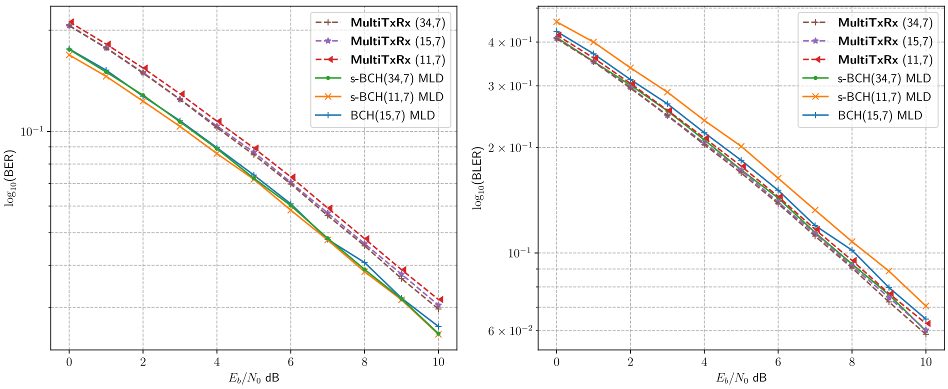


Fig. 13. BER and BLER in the Block Fading channel for MultiTxRx model with K = 7 bits and N = [11, 15, 34] compared with shortened BCH codes s-BCH(11,7), s-BCH(34,7) and BCH code (15,7) maximum likelihood decoding (MLD). The proposed module was originally trained on the AWGN channel and is not trained for the Block Fading channel.

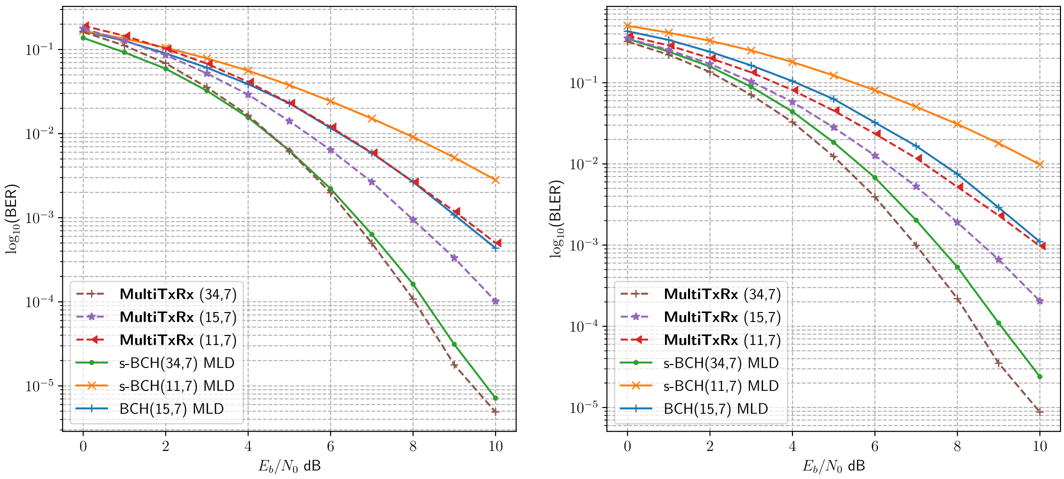


Fig. 14. BER and BLER in the bit fading channel for MultiTxRx model with K = 7 bits and N = [11, 15, 34] compared with shortened BCH codes s-BCH(11,7), s-BCH(34,7) and BCH code (15,7) maximum likelihood decoding (MLD). The proposed module was originally trained on the AWGN channel and is not trained for the bit fading channel.

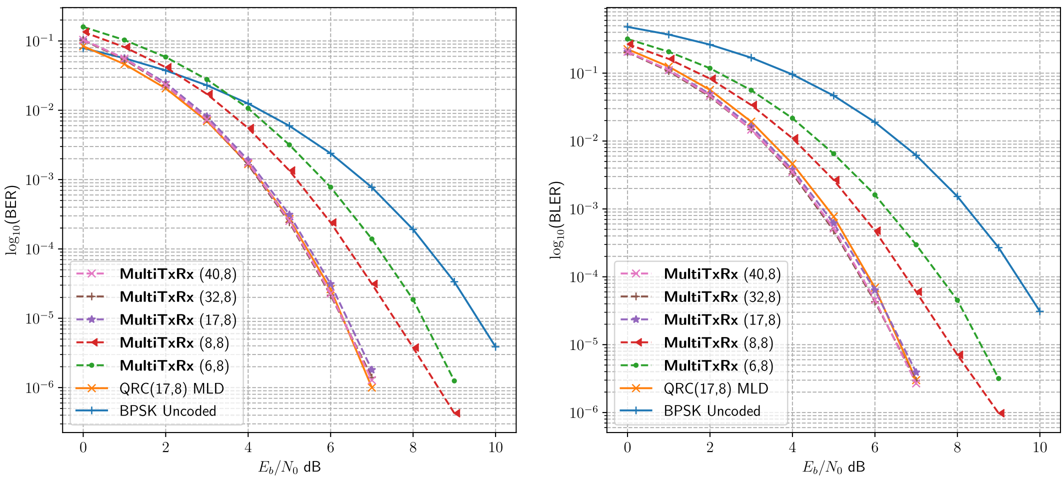


Fig. 15. BER and BLER in the AWGN channel for MultiTxRx model with K = 8 bits and N = [6, 8, 17, 32, 40] compared with BPSK uncoded and Quadratic Residue Code (QRC) K = 8, N = 17 maximum likelihood decoding (MLD), and trained with random SNR. The code (6,8) provides a higher channel usage than uncoded BPSK at 1.33 bits per channel usage.

rates is similar to the baseline code QRC(17,8) and BLER is slightly lower. The BER in the Block Fading channel, shown in Fig. 16, is worse than the target baseline QRC code, however, as we have seen in the other configurations, the BLER for the same code size and lower code rates is slightly better than the reference code. In the Bit Fading channel, both BER and BLER achieve equal or better performance than the reference QRC(17,8) code. However, the BER for higher code rates $K = 8, N = 8$ and $K = 8, N = 6$ do not perform as well as the uncoded BPSK in lower SNR, but do achieve gains for the BLER. This is also apparent in the AWGN channel.

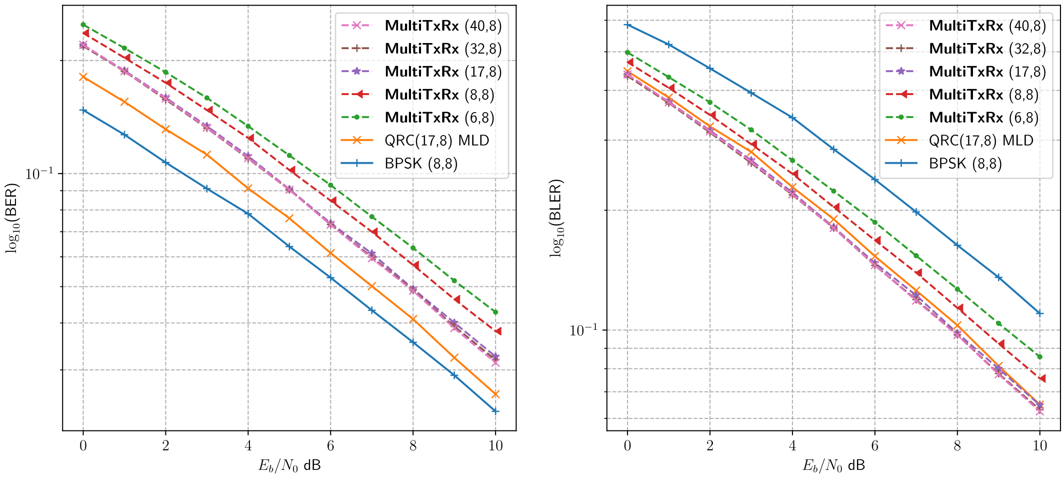


Fig. 16. BER and BLER in the Block Fading channel for MultiTxRx model with K = 8 bits and N = [6, 8, 17, 32, 40] compared with BPSK uncoded and Quadratic Residue Code (QRC) K = 8, N = 17 maximum likelihood decoding (MLD). The proposed module was originally trained on the AWGN channel and is not trained for the Block Fading channel.

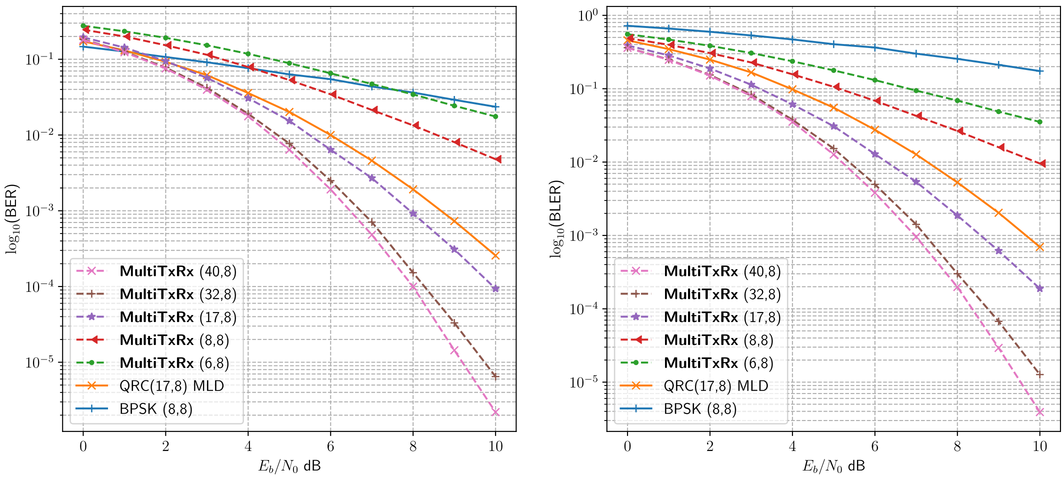


Fig. 17. BER and BLER in the bit fading channel for MultiTxRx model with K = 8 bits and N = [6, 8, 17, 32, 40] compared with BPSK uncoded and Quadratic Residue Code (QRC) K = 8, N = 17 maximum likelihood decoding (MLD). The proposed module was originally trained on the AWGN channel and is not trained for the bit fading channel.

5. Discussion

Comparison of the proposed model with the selected codes s-BCH(11,7), BCH(15,7), s-BCH(34,7) and QRC(17,8), demonstrated lower BLER in each of the channels and notably under the Bit Fading channel without retraining. While the BLER was close to the extended Hamming(8,4) code in each of the channels. In both the AWGN and Block Fading channels the BER was often poorer than the comparison code. As noted this is due to the classification for an entire code word rather than at the bit level and for the Block Fading channel, this effect of fading can be mitigated through the use of bit interleaving. However, the performance of a code is also dependent on the smallest minimum distance between all code words. Since the transmitter learns continuous codes, instead of binary codes, the minimum Euclidean distance is more appropriate measure of distance for those codes.

Fig. 18 shows the Euclidean distances between each of the learnt code words in the $K = 4, N = 8$ code. Ideally the transmitter should learn a constellation related to the distance between messages. In some cases, there is a larger distance between message code words with a message Hamming distance of 1, than those message code words with a larger message Hamming distance. For example, the Euclidean distance between code words for messages 0000 and 0001, a Hamming distance of 1, is larger than the Euclidean distance between code words for messages 0000 and 0111 with a Hamming distance of 3. The confusion matrix for the classifier is shown in Fig. 19, for messages 0000 and 0111 the percentage of incorrect classifications is approximately 3%, slightly higher than the incorrect classification between 0000 and 0001. The minimum Euclidean distance of the code does appear to be related to the performance of the learnt code. For those codes which have a lower BLER than the comparative code, the minimum Euclidean distance and mean Euclidean distances are close to or exceed that of the corresponding code. Table 4 lists the minimum, mean and variance of the Euclidean distance d_E calculated for the constellations of the learnt and comparison codes.

The changes to the AE to support multiple code rates does require an increase in the number of parameters overall within the neural network. This is to support generalisation over multiple code rates. However, the use of a common shared path for multiple codes does reduce the total number of parameters required in comparison to training separate models. The size of four single AE neural networks are shown in Table 5. The proposed branching model requires less parameters in a branching AE that can produce four different code rates in comparison to four separate AE.

The proposed models produced gains in BLER in comparison to the conventional codes under each of the channels. However it is not clear whether to attribute this gain to the learnt code or the inference supported by the AE. To investigate this, we developed a table based transmitter and MLD receiver for the code rate $K = 7, N = 15$. Symbols

Table 4
Computed minimum, mean and variance of euclidean distances for learnt and BPSK modulated reference codes.

Code rate	$d_{E_{min}}$	$E[d_E]$	$Var[d_E]$	Reference code	Code $d_{E_{min}}$	Code $E[d_E]$	Code $Var[d_E]$
K = 4, N = 8	3.83	4.12	0.08	Ext Hamming(8,4)	4	4.11	0.17
K = 7, N = 11	3.67	4.69	0.2	s-BCH(11,7)	3.46	4.66	0.47
K = 7, N = 15	4.38	4.38	0.19	BCH(15,7)	4.47	5.46	0.44
K = 7, N = 34	6.91	8.3	0.29	s-BCH(34,7)	6.63	8.25	0.53
K = 8, N = 17	4.34	5.82	0.23	QRC(17,8)	4.9	5.8	0.47

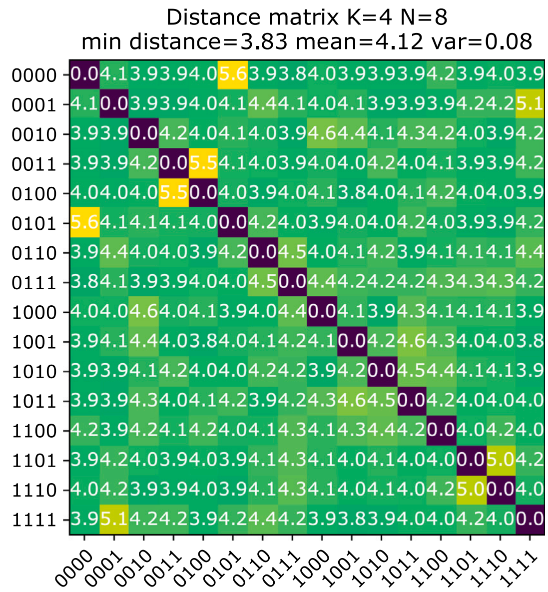


Fig. 18. Euclidean distances between pairwise codewords for each input sequence, learnt by the K = 4, N = 8 MultiTxRx auto-encoder.

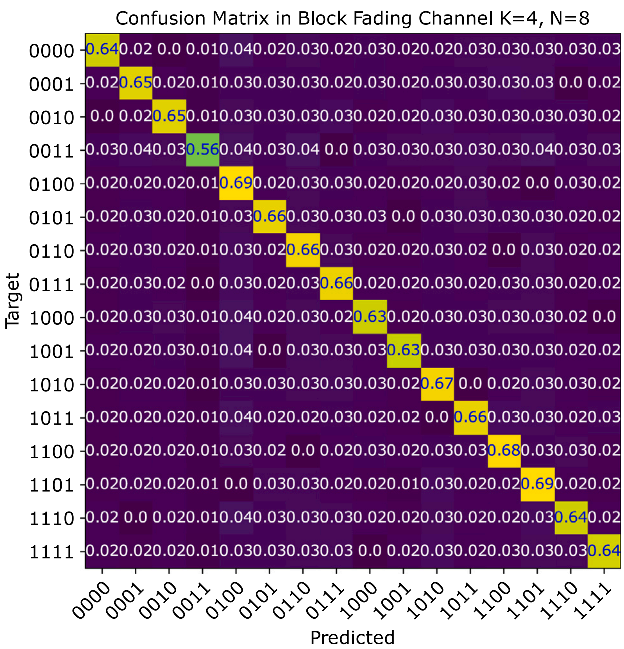


Fig. 19. The confusion matrix for the K = 4, N = 8 code rate under the block fading channel. Figures are relative to the predicted labels. While the classifier achieves a high level of accuracy on the BLER, there is sufficient difference between messages to cause high BER.

for corresponding 7 bit messages output by the *MultiTxRx* K = 7, N = 15 model were stored in a lookup table and transmitted over a AWGN channel. If the gain was solely due to the learnt receiver, we

Table 5
The number of parameters for combined separate code rate models versus the multi-task shared path model. The shared path architecture provides less total parameters than separate models for each code rate.

Model variant	K	N	Parameters
K = 4 N = 4 bit model	4	$i \in \{4\}$	8711
K = 4 N = 8 bit model	4	$i \in \{8\}$	9951
K = 4 N = 16 bit model	4	$i \in \{16\}$	12 431
K = 4 N = 20 bit model	4	$i \in \{20\}$	13 671
Total			44 764
4 bit multi-rate model	4	$i \in \{4, 8, 16, 20\}$	28 359

expect the MLD receiver to exhibit higher BLER. The MLD receiver performed nearest neighbour decoding for received channel values against the table of modulated symbols. The performance of the MLD receiver matched the performance of the proposed branching AE model in the corresponding channel (Fig. 20). This indicates that the gains observed are generated due to the learnt constellations resulting from training. This approach demonstrates potential use of DL for wireless communications as a method for code design which may be applied independently of the trained model.

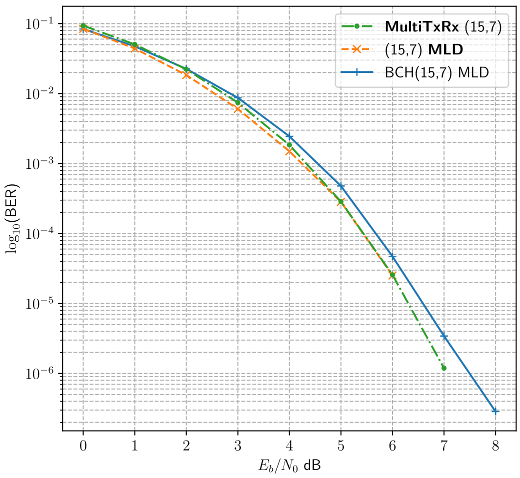
6. Conclusions and future work

In this article we have presented a branching AE architecture capable of automatically learning multiple code rates for AMC schemes. The proposed branching architecture extends applications of the AE architecture beyond the learning of a single code rate to the learning of multiple code rates. The choice of assumed channel during training is highly influential to the resulting performance of the AE in other channels. As a result, the ability to train a receiver separately on a real channel provides the ability to further optimise the system performance after deployment. The proposed branching AE for multiple code rates, is demonstrated to perform well under a variety of changing channel conditions, achieving gains in BLER compared to the selected conventional codes. By leveraging an AMC scheme the approach offers the potential to mitigate the requirement of receiver tuning in AE for wireless communications. However, there remains a number of limitations for the practical application of the DL approach requiring further investigation.

First, in this article we have assumed perfect synchronisation at the receiver. While it is possible to apply conventional methods for synchronisation with learnt modulation and coding schemes, it is desirable that synchronisation be addressed as part of the end-to-end AE architecture.

Second, classification based architectures not only do not scale to higher message lengths, but cannot provide error correction functionality. Hence work on bit-wise decoding for longer message lengths, either as part of a concatenated code, or as a standalone network will be a significant part of the practical application of such models, some of this work has already been described in the related work section of this paper.

Third, there has been work investigating the sensitivity of such architectures to their training conditions and whether they are brittle in terms of adversarial attacks. While we do not directly explore this concern, there is a connection between network regularisation and training methods required to mitigate adversarial attacks. In [29]



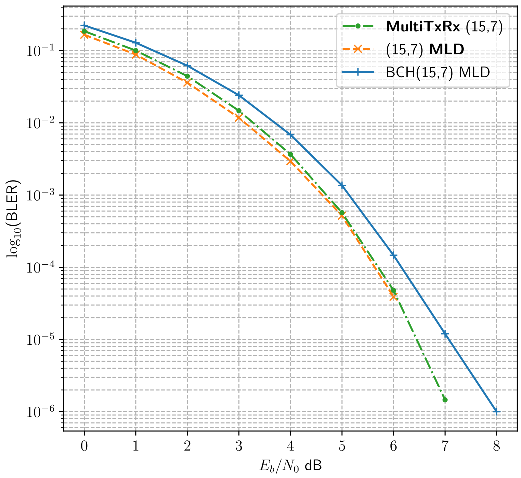


Fig. 20. BER and BLER in the AWGN channel of the learnt constellation for the (15,7) code produced by a table based transmitter and MLD receiver compared with the end-to-end model and BCH(15,7) code.

conventional Hamming codes are shown to be more robust under adversarial and jamming attacks than AEs. It is suggested in [30] that adversarial examples are transferable across different models, thereby enabling black-box attacks. This raises the importance for future investigation into regularisation methods for end-to-end learning in wireless communications and evaluation under adversarial interference.

Fourth, as we have discussed, the Euclidean distances between messages for neighbouring codes may be larger those several message bits away. This negatively impacts the performance of the BER, as misclassification results in a higher number of incorrect bits. Future work should investigate the ability of the transmitter to learn distance based relationships between source messages. In addition, while we have assumed no channel information at the receiver, it may be possible to incorporate or learn such information to enhance receiver performance in the end-to-end learning scenario.

The tuning of the receiver over varying channel conditions would be time consuming in a deployed system and may lead to poor performance on the original channel, for which it was first trained. Whether it is practical to tune a receiver over the air, and how much training is required, is a matter for consideration. A practical solution may be to use a branching AE with multiple code rates under changing conditions. This would permit operation whilst a separate model is adapted in the background. The question of how to update such a model while mitigating catastrophic forgetting in changing channel conditions deserves further investigation.

Finally, the mapping between channel environment and choice of code rate relies on measurements such as expected BER and BLER over associated SNR. It is feasible to imagine the joint learning of AMC and channel performance mapping, extending the work described in [31,32]. More recent research in the industrial internet of things (IIoT) consider wireless communications as part of a joint optimisation objective in seeking to reduce energy consumption over the collective sensor network [33,34]. A potential application would be to learn energy efficient communication schemes for the IIoT setting, which are adaptive to operational constraints in addition to channel conditions, in an end-to-end manner.

The flexibility of the AE architecture provides competitive performance not only in learning a single code rate, but also as we have shown, in learning AMC schemes with varying error-rate performance and spectral efficiencies. By framing the learning problem as MTL, the proposed architecture enables the deployment of a single model, instead of requiring multiple separate models for each code rate.

CRediT authorship contribution statement

Christopher P. Davey: Conceptualisation, Methodology, Implementation. **Ismail Shakeel:** Conceptualisation, Coordination, Editorial. **Ravinesh C. Deo:** Coordination, Editorial. **Ekta Sharma:** Editorial. **Sancho Salcedo-Sanz:** Coordination, Editorial. **Jeffrey Soar:** Coordination, Editorial.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data used in this article is generated through simulation. The simulation process is described in the methodology section of the article, alongside the energy constraints and channel distortions.

Acknowledgements

This research is supported by UniSQ-DSTG Postgraduate Research Scholarship 2021–2024 on the ‘Design of Efficient Artificial Intelligence Algorithms for Future Communication Systems’. It is funded by the Department of Defence, Commonwealth of Australia under a DSP Scholarship (Project-Based) Agreement 10254.

References

[1] G. Caire, K.R. Kumar, Information theoretic foundations of adaptive coded modulation, Proc. IEEE 95 (12) (2007) 2274–2298.
[2] Timothy O’Shea, Jakob Hoydis, An introduction to deep learning for the physical layer, IEEE Trans. Cogn. Commun. Netw. 3 (4) (2017) 563–575.
[3] S. Dörner, S. Cammerer, J. Hoydis, S. t. Brink, Deep learning based communication over the air, IEEE J. Sel. Top. Sign. Proces. 12 (1) (2018) 132–143.
[4] Hao Ye, Le Liang, Geoffrey Ye Li, Biing-Hwang Juang, Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels, IEEE Trans. Wirel. Commun. 19 (5) (2020) 3133–3143.
[5] F.A. Aoudia, J. Hoydis, End-to-end learning of communications systems without a channel model, in: 2018 52nd Asilomar Conference on Signals, Systems, and Computers, 2018, pp. 298–303.
[6] Michael Crawshaw, Multi-task learning with deep neural networks: A survey, 2020, arXiv preprint arXiv:2009.09796.
[7] Michael McCloskey, Neal J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of Learning and Motivation, Vol. 24, Elsevier, 1989, pp. 109–165.

[8] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, Yoshua Bengio, An empirical investigation of catastrophic forgetting in gradient-based neural networks, 2013, arXiv preprint arXiv:1312.6211.

[9] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, Andrew Gordon Wilson, Averaging weights leads to wider optima and better generalization, 2018, arXiv preprint arXiv:1803.05407.

[10] Fayçal Ait Aoudia, Jakob Hoydis, Model-free training of end-to-end communication systems, IEEE J. Sel. Areas Commun. 37 (11) (2019) 2503–2516.

[11] S. Cammerer, F.A. Aoudia, S. Dörner, M. Stark, J. Hoydis, S. ten Brink, Trainable communication systems: Concepts and prototype, IEEE Trans. Commun. 68 (9) (2020) 5489–5503.

[12] N.A. Letizia, A.M. Tonello, Capacity-driven autoencoders for communications, IEEE Open J. Commun. Soc. 2 (2021) 1366–1378.

[13] Rich Caruana, Multitask Learning (Ph.D. thesis), Carnegie Mellon University, Pittsburgh, PA, 1998.

[14] Kevis-Kokitsi Maninis, Ilija Radosavovic, Iasonas Kokkinos, Attentive single-tasking of multiple tasks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1851–1860.

[15] E. Armanious, D.D. Falconer, H. Yanikomeroglu, Adaptive modulation, adaptive coding, and power control for fixed cellular broadband wireless systems: some new insights, in: 2003 IEEE Wireless Communications and Networking, 2003, WCNC 2003, Vol. 1, 2003, pp. 238–242, vol.1.

[16] Joseph Downey, Dale Mortensen, Michael Evans, Janette Briones, Nicholas Tollis, Adaptive coding and modulation experiment using NASA's space communication and navigation testbed, in: 2016 Communications Satellite Systems Conference, ICSSC, 2016.

[17] Intae Hwang, Taewon Jang, Mingoo Kang, Sangmin No, Jungyoung Son, Daesik Hong, Changeon Kang, Performance analysis of adaptive modulation and coding combined with transmit diversity in next generation mobile communication systems, Future Gener. Comput. Syst. 20 (2) (2004) 189–196.

[18] D. Wu, S. Ci, Cross-layer design for combining adaptive modulation and coding with hybrid ARQ to enhance spectral efficiency, in: 2006 3rd International Conference on Broadband Communications, Networks and Systems, 2006, pp. 1–6.

[19] A.J. Goldsmith, S.G. Chua, Adaptive coded modulation for fading channels, IEEE Trans. Commun. 46 (5) (1998) 595–602.

[20] Shu Lin, Juane Li, Fundamentals of Classical and Modern Error-Correcting Codes, Cambridge University Press, Cambridge, 2021, URL https://www.cambridge.org/core/books/fundamentals-of-classical-and-modern-errorcorrecting-codes/19A81ED5D7E9C6A1EBB9657683B6E39C.

[21] Florence Jessie MacWilliams, Neil James Alexander Sloane, The Theory of Error-Correcting Codes, vol. 16, Elsevier, 1977.

[22] Raj Chandra Bose, Dwijendra K. Ray-Chaudhuri, On a class of error correcting binary group codes, Inf. Control 3 (1) (1960) 68–79.

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778.

[24] Sergey Ioffe, Christian Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Francis Bach, David Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 37, PMLR, 2015, pp. 448–456, URL https://proceedings.mlr.press/v37/ioffe15.html.

[25] Xavier Glorot, Antoine Bordes, Yoshua Bengio, Deep sparse rectifier neural networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[26] Prajit Ramachandran, Barret Zoph, Quoc V. Le, Searching for activation functions, 2017, arXiv preprint arXiv:1710.05941.

[27] L.N. Smith, Cyclical learning rates for training neural networks, in: 2017 IEEE Winter Conference on Applications of Computer Vision, WACV, 2017, pp. 464–472.

[28] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[29] M. Sadeghi, E.G. Larsson, Physical adversarial attacks against end-to-end autoencoder communication systems, IEEE Commun. Lett. 23 (5) (2019) 847–850.

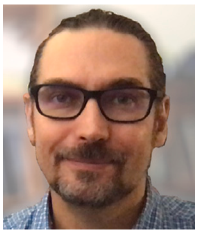
[30] Shan Ai, Arthur Sandor Voundi Koe, Teng Huang, Adversarial perturbation in remote sensing image recognition, Appl. Soft Comput. 105 (2021) 107252, URL https://www.sciencedirect.com/science/article/pii/S1568494621001757.

[31] S. Kojima, K. Maruta, C.J. Ahn, Adaptive modulation and coding using neural network based SNR estimation, IEEE Access 7 (2019) 183545–183553.

[32] P.V.R. Ferreira, R. Paffenroth, A.M. Wyglinski, T.M. Hackett, S.G. Bilen, R.C. Reinhart, D.J. Mortensen, Reinforcement learning for satellite communications: From LEO to deep space operations, IEEE Commun. Mag. 57 (5) (2019) 70–75.

[33] Sarogini Grace Pease, Russell Trueman, Callum Davies, Jude Grosberg, Kai Hin Yau, Navjot Kaur, Paul Conway, Andrew West, An intelligent real-time cyber-physical toolset for energy and process prediction and optimisation in the future industrial internet of things, Future Gener. Comput. Syst. 79 (2018) 815–829, URL https://www.sciencedirect.com/science/article/pii/S0167739X1630382X.

[34] Jiwei Huang, Han Gao, Shaohua Wan, Ying Chen, Aoi-aware energy control and computation offloading for industrial IoT, Future Gener. Comput. Syst. 139 (2023) 29–37, URL https://www.sciencedirect.com/science/article/pii/S0167739X22002916.



Christopher P. Davey has a Master of Information Technology degree from the Queensland University of Technology (QUT, Australia) in 2007 and completed a Master of Science majoring in mathematics and statistics from The University of Southern Queensland (UniSQ, Australia) in 2020. Chris has over a decade of professional experience in software development and systems integration. He is currently progressing work on a PhD program at UniSQ with the focus of his research being on "Deep Learning for Wireless Communications". He has worked on "Artificial Intelligence for Decision-Making (AI4DM)" and "AI-enabled communicating systems" research project funded by the Australian Government's Department of Defence.

Ismail Shakeel received a Ph.D. degree in Telecommunications in 2007 and a BEng (Hons) degree in Electronic Engineering in 1997 from the University of South Australia. He has completed two master's degrees at the University of Canterbury (NZ) and Monash University in 2001 and 2002 respectively. Ismail joined Defence Science and Technology Group (DSTG) in 2011 and is currently with the Information Sciences Division at DSTG. Ismail is also an Adjunct Professor at the University of Southern Queensland. Before joining DSTG, he has worked in both academia and industry and holds a patent and generated more than 40 technical reports and publications in the field of telecommunications. Ismail's current research interests include signal detection and classification techniques, artificial intelligence-enabled wireless communication, interference-resistant signalling, chaotic communication, and cooperative wireless communication.

Ravinesh C. Deo is a Highly Cited Author, 2021 Clarivate) leads UniSQ's Advanced Data Analytics Lab as Professor at the University of Southern Queensland (UniSQ), Australia. He is a Clarivate Highly Cited Researcher with publications ranking in top 1% by citations for field and publication year in the Web of Science citation index and is among scientists and social scientists who have demonstrated significant broad influence, reflected in the publication of multiple papers frequently cited by their peers. He leads cross-disciplinary research in deep learning and artificial intelligence, supervising 20+ Ph.D./M.Sc. Degrees. He has received Employee Excellence Awards, Elsevier Highly Cited Paper Awards, and Publication Excellence and Teaching Commendations. He has published more than 270 articles, 150 journals, and seven books with a cumulative citation that exceeds 11,600.

Ekta Sharma holds a Ph.D. degree in Artificial Intelligence from University of Southern Queensland, Australia. She completed her M.Phil., M.Sc. (Operations Research), and B.Sc. (Mathematical Sciences), from the University of Delhi, India. She has over a decade of experience in both Academia and Industry across varied roles from Area Manager to Learning Advisor, Lecturer, and Researcher at Universities in Europe, India, and Australia. Her research work has received funding from the Australian Government, the Australian Defence Science and Technology Group, The Australian Mathematical Sciences Institute, and the Australian Tropical Agriculture Institute. She is working on varied cross-disciplinary research projects in artificial intelligence and Wireless Communications.

Sancho Salcedo-Sanz was born in Madrid, Spain, in 1974. He received the B.S degree in Physics from Universidad Complutense de Madrid, Spain, in 1998, the Ph.D. degree in Telecommunications Engineering from the Universidad Carlos III de Madrid, Spain, in 2002, and the Ph.D. degree in Physics from Universidad Complutense de Madrid in 2019. He spent one year in the School of Computer Science, The University of Birmingham, U.K, as postdoctoral Research Fellow. Currently, he is a Full Professor at the department of Signal Processing and Communications, Universidad de Alcalá, Spain. He has co-authored more than 240 international journal papers in the field of Machine Learning and Soft-Computing and its applications. His current interests deal with Soft-computing techniques, hybrid algorithms and neural networks in different problems of Science and Technology.

Jeffrey Soar is Personal Chair in Human-Centered Technology at the School of Business, University of Southern Queensland. His research is in AI, e-business, e-health, technology and development, and social and organisational change. He came to academic research from a long and distinguished career in industry including as chief information officer in government agencies in Australia and New Zealand.

5.3 Links and Implications

This article proposed a multi-task learning architecture for AMC schemes and demonstrated that the model was able to match or outperform conventional codes in the Rayleigh fading channel environment for block and bit fading, while only having trained in the AWGN channel environment. By providing a single DL architecture, the approach simplifies the task of E2E learning for AMC schemes. The proposed approach enables future potential for the joint learning of both coded modulation combined with a control algorithm such as DL based RL, which is able to learn when to select a new coded modulation scheme based on the channel conditions. While OAL methods adapt DL based communications systems to the true channel environment, their key disadvantage is the training time required to develop the system over-the-air. By training a DL wireless communications system that is capable of AMC it is possible to provide adaptive communication schemes that are reliable and performant under a variety of channel environments without the need for retraining.

Chapter 6: Paper 3 - Channel Agnostic Training of Transmitter and Receiver for Wireless Communications

6.1 Introduction

This chapter presents a copy of the article published in Sensors (vol. 23 (2023), no. 24, p2848, ISSN: 1424-8220, doi:10.3390/s23249848. `https://www.mdpi.com/1424-8220/23/24/9848`).

Training DL based wireless communications systems OAL must approach backpropagation in such a way that either gradients between receiver and transmitter are approximated or that the channel environment is approximated to then enable E2E learning. This article investigates the process of OAL training of a transmitter and remote receiver where the transmitter is jointly optimised E2E with a local receiver which learns to imitate the estimates produced by the remote receiver. The estimates for the remote receiver are provided through a reliable feedback channel. The novelty of the approach proposes that the errors produced in the remote receiver estimates contains implicit knowledge of the channel environment, and that learning to imitate these estimates enables the transmitter to learn a modulation that is optimised for that environment. This is demonstrated through training the proposed system under multiple channel environments and comparing to methods such as Receiver tuning as well as training without imitation.

Research Highlights

- The article proposes a novel over-the-air training method and develops machine learning enabled coding and modulation schemes for the transmitter and the receiver without an assumed channel model.
- It develops a Disjoint Learning method that uses a transmitter side (local) channel/receiver to imitate the learning process of the remote receiver and enable supervised learning of the transmitter through backpropagation.

- The performance of the proposed Disjoint Learning method is demonstrated to be equivalent or better than the fully connected architecture.
- The proposed method is shown to achieve significant performance improvements against the Receiver Tuning training method.

6.2 Published Article 3

sensors

MDPI

Article

Channel-Agnostic Training of Transmitter and Receiver for Wireless Communications

Christopher P. Davey [1], Ismail Shakeel [2], Ravinesh C. Deo [1,*] and Sancho Salcedo-Sanz [1,3]

[1] School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, QLD 4300, Australia; chris.davey@unisq.edu.au (C.P.D.); sancho.salcedo@uah.es (S.S.-S.)
[2] Spectrum Warfare Branch, Information Sciences Division, Defence Science and Technology Group (DSTG), Edinburgh, SA 5111, Australia; ismail.shakeel@defence.gov.au
[3] Department of Signal Processing and Communications, Universidad de Alcalá, Alcalá de Henares, 28805 Madrid, Spain
* Correspondence: ravinesh.deo@unisq.edu.au

Abstract: Wireless communications systems are traditionally designed by independently optimising signal processing functions based on a mathematical model. Deep learning-enabled communications have demonstrated end-to-end design by jointly optimising all components with respect to the communications environment. In the end-to-end approach, an assumed channel model is necessary to support training of the transmitter and receiver. This limitation has motivated recent work on over-the-air training to explore disjoint training for the transmitter and receiver without an assumed channel. These methods approximate the channel through a generative adversarial model or perform gradient approximation through reinforcement learning or similar methods. However, the generative adversarial model adds complexity by requiring an additional discriminator during training, while reinforcement learning methods require multiple forward passes to approximate the gradient and are sensitive to high variance in the error signal. A third, collaborative agent-based approach relies on an echo protocol to conduct training without channel assumptions. However, the coordination between agents increases the complexity and channel usage during training. In this article, we propose a simpler approach for disjoint training in which a local receiver model approximates the remote receiver model and is used to train the local transmitter. This simplified approach performs well under several different channel conditions, has equivalent performance to end-to-end training, and is well suited to adaptation to changing channel environments.

Keywords: deep learning; channel free training; wireless communications; over-the-air training; neural networks

Citation: Davey, C.P.; Shakeel, I.; Deo, R.C.; Salcedo-Sanz, S. Channel-Agnostic Training of Transmitter and Receiver for Wireless Communications. *Sensors* **2023**, *23*, 9848. https://doi.org/10.3390/s23249848

Academic Editor: Kenneth Brown

Received: 24 November 2023
Revised: 11 December 2023
Accepted: 13 December 2023
Published: 15 December 2023

1. Introduction

The primary goal of a wireless communications system is to transmit a message over-the-air (the channel environment) to a receiver such that the message can be recovered without error. However, the channel environment causes distortions in the transmitted signal that impede perfect recovery of the message. To improve message recovery, communications systems are designed with multiple signal processing blocks and with complementary components between the transmitter and receiver for each stage (coding/decoding, modulation/demodulation, filtering/detection). Figure 1 illustrates a simple wireless communications system comprising a transmitter, channel, and receiver. Each of these stages is traditionally designed and optimised independently while assuming a fixed mathematical model of the channel.

More recently, deep learning (DL) in wireless communications systems has been applied to jointly optimise functions for the transmitter and receiver over an assumed channel model [1]. Such approaches offer an alternative to the block design of communications systems, and may achieve better performance in complex channels without a formal model [1]. The supervised learning procedure enables the transmitter to learn complex domain symbols, thereby maximising the ability of the receiver to de-noise and map soft channel outputs to the original message. The DL auto-encoder (AE) architecture is a proven approach for application to automatic feature learning, and is coupled with noise distortions during learning to enable the decoder component of the architecture to learn robust features for de-noising and estimation [2].

During training, the perturbations provided by an assumed channel model help the transmitter (encoder) to learn robust features through the process of backpropagation. Backpropagation communicates the loss at the receiver (decoder) by applying the chain rule with respect to the training loss function, which requires a differentiable channel function to pass the gradients from the receiver to the transmitter. The true channel environment prevents backpropagation between the transmitter and receiver, representing a key challenge in the over-the-air training of AE for wireless communications systems.

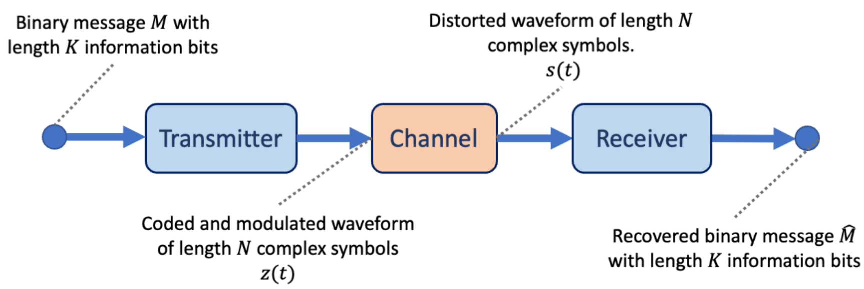


Figure 1. A simplified wireless communications system comprising a transmitter, a channel environment, and a receiver. The transmitter takes the input message block, then performs encoding and modulation prior to sending it over the channel. The channel distorts the waveform; such distortions can include noise and fading. The receiver must detect and filter the content of the received waveform, then demodulate and decode the data in order to recover the original message.

Research into over-the-air learning for wireless communications systems has demonstrated approaches in which the transmitter and receiver can be trained in a disjoint manner. DL approaches which leverage the AE architecture to model the transmitter, channel, and receiver have approached the problem by training an end-to-end system offline with an assumed channel model (Joint Learning) and tuning the receiver model online against the pretrained transmitter [3] (Receiver Tuning). During the tuning phase, the transmitter is not updated under the true channel conditions, preventing improvement of the code learned by the transmitter during the tuning phase. Thus, any improvement under the new channel depends on the adaptation of the receiver.

The transmitter learns a code that relies on the properties of the channel environment, which are modelled during training. The Joint Learning process results in a code that maximises the mutual information between the transmitted (channel input) and received (channel output) symbols through direct observation of the channel [4]. In contrast, conventional coding methods counteract channel effects such as fading by introducing redundant symbols (diversity) or using estimates of fading coefficients (channel state information) for precoding at the transmitter or correction at the receiver [5]. DL techniques have demonstrated the ability to learn accurate estimates for channel state information, and have been applied to correction and signal detection at the receiver [6,7]. The application of DL to channel modelling has led to the adoption of generative adversarial network (GAN), which can learn to emulate the stochastic channel environment [8], motivating the potential application of DL to either explicitly model the channel environment or implicitly extract channel state information during over-the-air learning (OAL).

Two methods of extending DL to OAL involve feedback from the receiver to enable learning a proxy of the channel, thereby permitting backpropagation between the transmitter and channel model [9–12] (Channel Approximation). Another approach is Gradient approximation, in which the gradient at the transmitter is approximated through variants of finite difference approximation or reinforcement learning (policy-gradient learning) [13–16]. Additional methods involve multi-agent approaches such as Collaborative Agent Learning coordinated by specific training protocols [17], which is able to include a variety of learning algorithms other than DL. Both the channel and gradient approximation approaches have demonstrated equivalent performance to the end-to-end joint learning approach [11,13,14], while the Collaborative Agent Learning method has demonstrated performance close to conventional codes [17].

In this paper, we refer to Receiver Tuning, Channel Approximation and Gradient Approximation as methods of Disjoint Learning and regard these as separate from the Collaborative Agent Learning approach. We present an additional method of Disjoint Learning, "Learning through Imitation", that is situated between Channel Approximation and Gradient Approximation, where a local channel/receiver model is developed using estimates from the actual receiver to imitate the behaviour of the channel/receiver at the transmitter side. This enables the application of supervised learning to train the transmitter using backpropagation. This approach does not model the channel directly; instead, it learns to mimic the errors made by the remote receiver and acts as a proxy for the remote receiver model. We use simulation to produce equivalent results to the end-to-end Joint Learning approach first demonstrated in [1] and show that this method outperforms receiver tuning. To show that the local receiver model approximates the remote receiver model, we compare the process of learning without feedback to that of learning with feedback, and demonstrate that learning through imitation exceeds the performance of learning without feedback.

Therefore, with the aim of providing a novel method for channel-agnostic over-the-air training of both transmitter and receiver for resilient wireless communications, the primary objectives of this study are as follows:

- To propose a novel over-the-air training method and develop machine learning enabled coding and modulation schemes for the transmitter and the receiver without an assumed channel model.
- To develop a Disjoint Learning method that uses a transmitter-side (local) channel/receiver to imitate the learning process of the remote receiver and enable supervised learning of the transmitter through backpropagation.
- To demonstrate that the performance of the proposed Disjoint Learning method is equivalent or better than the fully connected architecture.
- To show that the proposed method achieves significant performance improvements against the Receiver Tuning training method.

The rest of this paper is organised in the following way: Section 2 provides a brief overview of related work; Section 3 describes our proposed model, training, and simulation methods; Section 4 presents results for the proposed method and provides a discussion of the results and modelling approach; and Section 5 draws conclusion and proposes future directions for investigation.

2. Background and Related Works

The canonical application of DL for the joint learning of a wireless communication system is presented in [1]. An AE transmitter and receiver model was shown to perform equivalently to short uncoded and Hamming(7,4) coded messages ($K = 4$ information bits and $N = 7$ code bits) on the Additive White Gaussian Noise (AWGN) channel [1]. The authors observed the relationship between the choice of energy constraint and constellation learned by the transmitter. The influence of the channel on the system was shown by training two pairs of transmitter and receiver AEs on an interference channel. The transmitters learned to counteract the interference channel by developing orthogonal codes [1]. It is acknowledged that both symbol-wise AE (classification mapping code word to message) and bit-wise AE (modelled as K-bit outputs) are limited in their application to smaller codes due to the dimensionality of a possible 2^K messages for K information bits. The joint learning approach demonstrates the inclusion of an assumed channel transfer function in the design of the network, and must be trained offline. This prevents joint optimisation on the true channel environment.

Receiver tuning, inspired by transfer learning, was carried out after the joint learning phase and used to update the trained receiver on the true channel in [3]. The resulting system was compared with differential quadratic phase shift keying (DPSK) in both simulated AWGN and over-the-air channels. The simulated channel included impairments for timing, phase, and frequency offsets, while the receiver model was developed to correct for these distortions before decoding [3]. Receiver tuning was demonstrated to improve the performance of the AE over the end-to-end model, but did not improve upon the DPSK modulation. However, the approach demonstrated a practical way forward in tuning AE models over-the-air. The primary disadvantage of receiver tuning is that the transmitter remains fixed during the tuning phase and does not adapt to the true channel distortions compensated by adaptation at the receiver.

Methods for disjoint learning emerged to address the limitations of receiver tuning and permit over-the-air training of both transmitter and receiver models. Channel approximation methods using GANs [18,19] have been applied to train a proxy for the channel in response to feedback and enable the transmitter to be trained with backpropagation through the generator channel model [9–12]. In [9], a channel model inspired by the GAN approach was trained to approximate the channel response directly, and the transmitter was updated by alternating backpropagation phases between channel and receiver loss. A local receiver (acting as the discriminator) is required in order to enable end-to-end learning for the transmitter, and leverages the channel model for backpropagation. This approach was extended in [10] to leverage a separate discriminator network, while a variational neural network was incorporated in [20] to describe the channel distribution in the generator. The variational method has been shown to better approximate the variance of the channel response for a range of channels in comparison to the previous method based on mean squared error (MSE) loss [10]. These approaches introduce a separate training procedure to train the generator in order to approximate the true channel environment.

In [21], a conditional GAN was trained to approximate the AWGN and Rayleigh fading channels conditioned on the pilot symbols in [11], then used to optimise a transmitter and receiver for symbol classification. The channel model was shown to approximate the AWGN perturbations for a quadrature amplitude modulation (QAM) of sixteen symbols [11]. The performance of the system was shown to be equivalent to a Hamming(7,4) code over AWGN and to perform similarly to coherent detection in a Rayleigh fading channel [11]. The approach was later combined with convolutional neural network (CNN) modules for bit-wise estimation for longer message lengths in [12]. A simple feed-forward GAN was compared with 4-QAM Hamming(7,4) code under AWGN. A CNN-GAN was compared to a convolutional code in the Rayleigh fading and selective-frequency channels in [12]. Performance in each channel was shown to be close to the conventional methods, and the importance of the pilot symbols was empirically demonstrated in the selective frequency channel [12]. The GAN approach introduces complexity to the training procedure due to the need to alternate between training the discriminator and generator as well as between the transmitter and receiver training phases.

A one-shot training approach for a conditional GAN was adopted in [22] to simplify the training procedure. It was used to train an AE model that supports longer messages lengths by combining the AE with bit-interleaved coded modulation (BICM) and an outer low-density parity-check (LDPC) code [22]. Comparison against a 16-QAM baseline and a AE-GAN trained on a simulated AWGN channel were made, as well as with a AE-GAN trained over-the-air and the reinforcement learning (RL)-based approach described in [16]. The AE-GAN trained on the true over-the-air channel environment demonstrated improved performance over the same approach trained on a simulated channel [22]. The approach required two stages, with the GAN first trained independently of the AE and later applied to train the AE on the receiver side. While it was suggested that the GAN framework could be used to model the channel without prior knowledge, the authors reported difficulties in training the GAN considering the presence of carrier frequency offset (CFO), which prevented the GAN from converging [22].

A separate channel model is not a necessity for optimisation of the transmitter and receiver models. Other approaches have focused on gradient approximation methods to support backpropagation at the transmitter. A finite difference gradient approximation method, Simultaneous Perturbation Stochastic Approximation (SPSA), was applied in [13]. The transmitter symbols were perturbed multiple times with a given noise distribution and the receiver errors were collected for each point and applied to approximate the gradient at the transmitter [13]. The model was demonstrated to be equivalent to uncoded quadrature phase-shift keying (QPSK) in AWGN and very close to theoretical uncoded QPSK in Rayleigh block fading channels. In addition, it was shown to be comparable in performance to the end-to-end AE described in [1]. The above process is computationally expensive. Results are taken from an average of 250 independent models; each time the gradient is approximated, the transmitter outputs are combined with a small perturbation vector and the receiver loss is calculated for each [13]. Due to the amount of sampling required to approximate the gradient, this method would encounter difficulty scaling to more complex transmitter models or longer message sequences.

An alternate gradient approximation approach proposed in [14] is based on policy gradient (PG) approximation. Such methods are applied in deep RL; an agent learns to exploit actions in response to the environment, resulting in the highest expected reward [23,24]. In [14], a penalty signal is provided by the receiver loss. The transmitter is trained to minimise the loss without an explicit model of the channel environment. Learning is achieved by alternating between the training of the receiver and the transmitter. This approach does not require a local proxy for the receiver, as the gradient can be estimated directly from the loss signal calculated for perturbations of the complex symbols learned at the transmitter. This process generates a stochastic sampling scheme equivalent to RL "policy" exploration [14]. The approach was evaluated in both AWGN and Rayleigh fading channels. In the latter, the receiver network was modified with a prior assumption of the channel distortion to learn estimates of the fading coefficients and reverse the fading prior to the discriminative layers of the network [14]. While the authors indicated that the training procedure requires more iterations than the end-to-end method, their evaluation demonstrated equivalent performance to end-to-end AE in both channels [14]. The method was tested over-the-air with software defined radio (SDR) in [15,16] and had a lower error rate in comparison to conventional codes. Both of these sources indicate that the method requires an extended training duration and that the variance of the receiver loss negatively impacts the convergence of the gradient at the transmitter [15,16]. To address the long training time, it has been proposed to pretrain the network offline and perform online tuning of both the transmitter and receiver [15].

The deep deterministic policy gradient (DDPG) approach was applied in [25] to address the issues around convergence described in [14] by applying both a "replay" buffer (sometimes termed an "experience" buffer) and a soft update rule used to transfer learned weights between a duplicate transmitter and an accompanying critic network. This method was reported to outperform the alternating algorithm in both Rayleigh and Rician fading channels [25]. The addition of the replay buffer requires additional memory to store previous receiver losses, and the additional critic network increases the complexity of the training algorithm in a trade-off with the improved learning at the transmitter.

The problem of training both the transmitter and receiver has been framed as a collaborative agent problem. These types of approaches are interesting because they can coordinate training between different types of learning algorithms for the transmitter and receiver. A hybrid approach called Collaborative Multi-Agent Learning was presented in [26]. This method trains a neural network transmitter using RL to learn the symbol constellation and a k-means clustering receiver to determine the number of symbols and estimate the message. A transmitter (Tx A) outputs a modulation for a given preamble, then transmits to a receiver (Rx B) over an AWGN channel, which produces an estimate of the message; this estimate is relayed through the second transmitter (Tx B) to a receiver on the originating side (Rx A), which is used to estimate a loss signal for the original transmitter (Tx A) [26]. This echo procedure has been shown to produce varying-order modulations under different training regimes for noise and energy constraints [26]. However, it did not achieve comparable results to the baseline QAM modulation [26]. The echo procedure is complex in that it requires two pairs of transmitter and receivers; each pair iteratively swaps between sending the original message to update each transmitter.

An echo protocol with a private preamble was applied in [17]. Pairs of collaborating agents share information about the learning task, and the difficulty of learning increases as less information is exchanged [17]. The authors asserted that their proposed echo protocol with private preamble enables learning of different types of agents and minimises the amount of information sharing between agents [17]. The method was demonstrated to perform similarly to QPSK under AWGN as well as in over-the-air experiments [17]. Both sources [17,26] leveraged a similar approach in defining transmitter and receiver pairs as agents during training, and both applied RL to train the transmitter. While neither approach outperforms conventional codes, the technique of using the receiver estimate as an echo is of interest for our method. Our proposed method learns to imitate the feedback from the remote receiver estimate, which includes the errors made during training.

Regularisation in DL seeks to reduce the bias of the network towards training data. It achieves this through reducing the complexity of the model during training [27]. Mechanisms include penalising weights (weight normalisation and averaging), perturbation of inputs (such as the transformations applied to images in computer vision), learning normalisation of activations (batch and layer normalisation), perturbation of network structure (such as drop-out), and training algorithms (such as stochastic gradient descent (SGD)). The use of incorrect labelling has been shown to provide regularisation for classification tasks [28]. This method makes use of a small noise rate to modify the ground-truth label of each class by selecting from weighted alternatives [28]. It had been shown to slow convergence and reduce overfitting of the model during training [28]. The authors used a fixed noise rate parameter and showed improvements when training reference models on several computer vision benchmarks [28]. While the noise rate is not decreased during training, this approach is relevant to our proposed method. Early in the learning process, the remote receiver yields a less accurate estimate which corresponds to a higher loss. The estimates become more accurate during the learning process, and the loss gradually decreases as learning progresses. The local channel/receiver is trained to imitate the estimates output by the remote receiver. In this manner, the learning process is comparable to training against noisy classification targets where the noise rate decreases over time. The purpose is to enable the local channel/receiver to learn from the noisy estimation process at the remote receiver.

The surveyed approaches for learning wireless communications systems have included joint learning, disjoint learning, and collaborative agent learning. Our focus is on joint and disjoint learning, with the the focus of this literature review on methods for training AE neural network models. Our proposed method differs from the GAN and RL methods surveyed above. In comparison to GAN methods, our method does not learn an explicit channel generator model and does not require a discriminator model during training. Instead, a local channel/receiver model is trained to imitate the remote receiver model. In comparison to RL-based methods, we do not perform gradient approximation; hence, we do not require multiple perturbations during the forward pass to estimate the gradient at the transmitter, and do not require additional support from methods such as a "replay" buffer to address variation in the loss estimate. Instead, the local channel/receiver model acts as a proxy for the remote receiver model to support end-to-end backpropagation at the transmitter. While we do leverage the remote receiver estimate as feedback, which is somewhat similar to the echo protocol in collaborative agent learning, we do not require additional coordination protocols for multiple agents and do not train transmitter/receiver pairs. Instead, our method trains a local proxy on the transmitter side against the feedback of estimates from the remote receiver. Our simplified approach removes the need for channel generative modelling, gradient approximation, or coordination protocols.

3. Methodology

In this section, we describe our proposed approach and the channel environment simulation used for training and evaluation of the resulting transmitter and receiver models. Section 3.1 briefly describes the communications system, Section 3.2 outlines our proposed method and training approach, and Section 3.3 describes the channel environments used for simulation during training and evaluation.

3.1. System Description

A wireless communications system aims to communicate a K bit message x. The transmitter converts the message into an optional code word of length N, and converts the message (or code) into a set of modulated symbols. It then combines the modulation with a carrier wave to transmit a set of complex values $z(t) \in \mathcal{C}$ with $t = 1 \ldots T$ timesteps and applies a filter to prevent inter-symbol interference. These values are transmitted through a channel environment that causes distortions including noise and fading effects. The channel environment is represented in our simulations as a channel transfer function $r(t) = h(z(t))$. The received signal $r(t)$ is filtered and imperfections are corrected, then it is demodulated and decoded to produce an estimate for the original K bit message y. In a wireless communications environment, there are mismatches at the transmitter and receiver in the timing, phase, and frequency between the transmitted signal $z(t)$ and received signal $r(t)$. Such imperfections can be simulated with the channel transfer function. However, for this work we assume perfect synchronisation and do not perform corrections for these offsets at the receiver, nor do we perform filtering at the transmitter and receiver. Our focus is on training a local transmitter DL model to perform modulation and coding, simulating the physical channel external to the DL models, and training the remote receiver DL model to estimate the original message.

3.2. Proposed Approach

We start with the joint learning of an end-to-end AE model, similar to the architecture described in [1]. This model, shown in Figure 2, consists of a transmitter neural network $Tx(x, \theta_t)$ with weights θ_t and a receiver neural network $Rx(r, \theta_r)$ with weights θ_r linked by an assumed channel function $h(z)$. The main paths of both networks consist of feed-forward dense modules followed by a rectified linear unit (ReLU) activation [29]. In the transmitter, a *tanh* activation is applied prior to an energy normalisation layer. The modelling approach focuses on small block codes using a symbol-wise representation; hence, input messages of K bits are one-hot encoded as 2^K words prior to presentation to the transmitter. A one-hot encoded vector x has length 2^K and contains a one at the index corresponding to the selected message and zeroes in all other index positions.

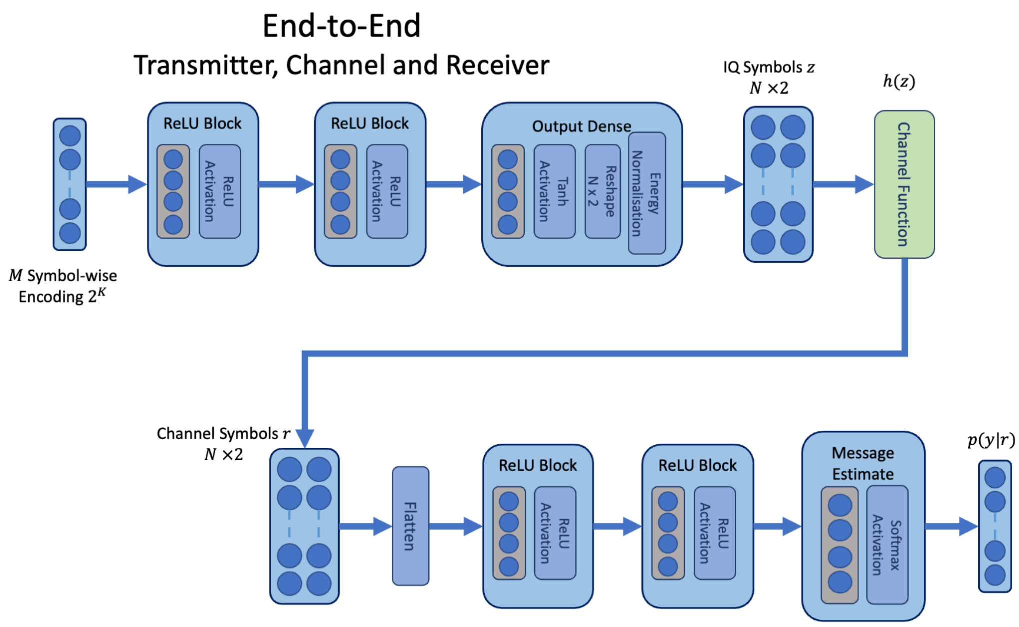


Figure 2. The end-to-end network architecture, where an assumed channel transfer function is defined as a layer within the network architecture.

A set of complex soft values $z \in \mathcal{C}$ are output by the transmitter neural network prior to the channel, and represent the code of length N; these are defined as two real numbered vectors for the in-phase and quadrature IQ coordinates within the network $z = Tx(x, \theta_t)$. Inputs to the receiver result from application of a channel transfer function $r = h(z)$, and the receiver converts these channel symbols back to a probability distribution over each message $p(y|r) = Rx(r, \theta_r)$. The probability vector contains an estimate for each of the possible 2^K messages using the *softmax* activation. This activation function takes as input the vector produced by the final dense layer f of the receiver neural network, which has 2^K linear units. The *softmax* activation is defined as $p(y_i|r) = \exp(f_i) / \sum_j^{2^K} \exp(f_j)$; the summation in the denominator ensures that the outputs sum to 1, and corresponds to a probability density. The model performs classification by taking the index with the highest probability as the index for the corresponding K bit word in the lookup table containing all possible words $\hat{M} = \arg\max p(y_i|r)$.

In the canonical AE, the transmitter model, assumed channel function, and receiver model are connected such that training can be carried out end-to-end with backpropagation. Backpropagation consists of a forward pass $p(y_i|r) = Rx(h(Tx(x, \theta_t)), \theta_r)$ and a backward pass that updates the weights at each layer by calculating the derivative with respect to the loss by application of the chain rule. The model is trained to minimise the cross-entropy loss between the true and estimated message labels in Equation (1). The expression $p(y_{true})$ indicates the target one-hot encoded vector for the true message presented during training. Typically, the network is presented with batches of data and the loss is averaged over the entire batch.

$$\mathcal{L}(p(y_{true}), p(y|r)) = -\sum_{i=1}^{2^K} p(y_i) \log p(y_i|r) \tag{1}$$

The backward pass calculates the gradients for the weights in the network. For the receiver, the backward pass applies the chain rule between the receiver network model and the loss function in Equation (2), and updates the weights by taking a small step in the direction of the gradient $\theta_r = \theta_r - \eta \nabla_{\theta_r}$, where η represents a small learning rate constant. For the transmitter, the backward pass includes the gradients from the receiver as well as the gradient for the channel function with respect to the transmitter model, and updates the transmitter weights accordingly: $\theta_t = \theta_t - \eta\nabla_{\theta_t}$. In stochastic gradient descent, the gradient is calculated over the batch and several enhancements to the method add features, for instance, momentum to dynamically control the step size during learning, or adaptive learning rates for different parameters of the model, such as in the Adam optimizer [30].

$$\nabla_{\theta_r} = \frac{\partial Rx(r,\theta_r)}{\partial\theta_r}\frac{\partial\mathcal{L}[p(y_{true}), Rx(r,\theta_r)]}{\partial Rx(r,\theta_r)} \tag{2}$$

$$\nabla_{\theta_t} = \frac{\partial Tx(x,\theta_t)}{\partial\theta_t}\frac{\partial h(Tx(x,\theta_t))}{\partial Tx(x,\theta_t)}\nabla_{\theta_r} \tag{3}$$

During receiver tuning, the transmitter and receiver models are detached from the channel layer and the receiver model is updated via backpropagation while the transmitter remains frozen. Therefore, receiver tuning does not require differentiation through the channel function.

The architecture we apply in disjoint learning consists of a disconnected transmitter model $Tx(x,\theta_t)$ and a receiver model $Rx(r,\theta_r)$, and we simulate a channel transfer function $h(z)$ separately from both models so that the channel does not participate in backpropagation. This is to simulate the process of over-the-air learning. In over-the-air learning, the channel may take on more complex behaviour than is captured by the assumed mathematical channel function. Therefore, training from the true channel is desirable, as it can permit the network to learn a coded modulation that is optimised for the true channel environment. As described in Section 2, the current approaches to disjoint learning achieve backpropagation at the transmitter by either explicitly learning the channel or by approximating the channel gradient. Instead of learning the channel directly, we rely on a local proxy for the remote receiver at the transmitter side, which we use to perform backpropagation without training an explicit channel model.

Before describing the training method, we first describe the structure of the network architectures for the transmitter and receiver models. The transmitter and receiver neural network architectures contain a series of fully connected dense blocks, similar to the end-to-end AE; however we add skip connections in the main path of each network. This architecture is illustrated in Figure 3. The skip connections, described as a “Skip Block” in the figure, assist backpropagation and combine features learned in the earlier hidden layers with the upper hidden layers [31]. In addition to the effect on backpropagation, skip connections are indicated to learn an ensemble of networks [32]. Each skip block is comprised of several dense blocks containing batch normalisation [33] and a nonlinear swish activation [34]. Input to and output from the transmitter follows the same principle as the end-to-end architecture, as does the input and output from the receiver. The layers, unit sizes, and groups within the transmitter are described in Table 1, while the receiver is described in Table 2. The dimension of the networks was arrived at through a manual process; while it is possible to use automated procedures for finding the best dimensions, such processes often tend to be computationally demanding and require a long duration. We chose a manual stepwise approach for simplicity, gradually increasing the dimensions of each layer by powers of 2. It is interesting to note that learning shorter codes appears to be more challenging than learning codes with longer lengths, requiring a larger dimension of the intermediate dense layer within the skip block for the 4/7 code rate as opposed to the uncoded 8 bit message.

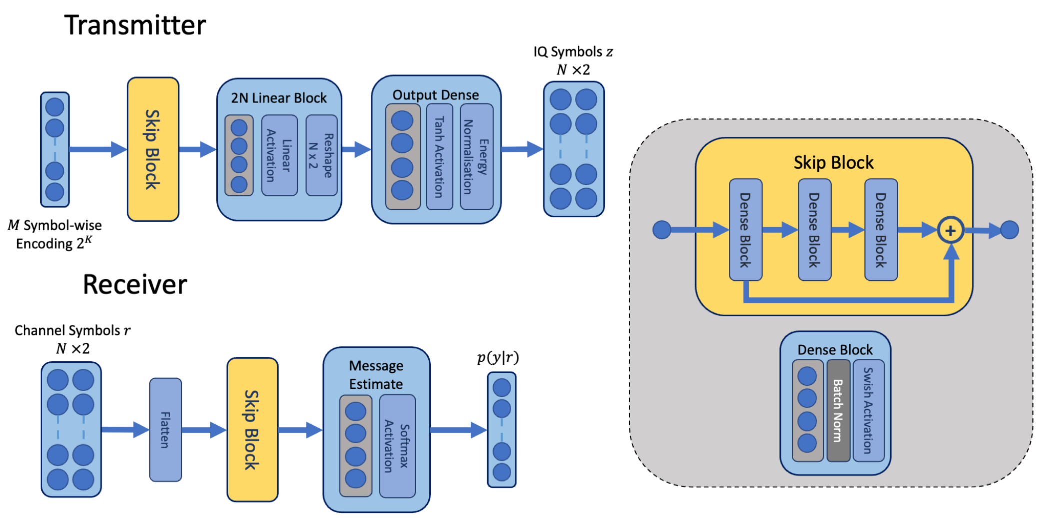


Figure 3. The architectural blocks for disjoint learning of transmitter and receiver. The same architecture is applied in both local and remote receivers. A channel transfer function is not assumed as part of the model.

Table 1. The transmitter consists of four groups: input, skip block, a linear transformation, and an output block. The number of units are specified for the dense layers, while batch normalisation and swish activation preserve the same dimension of output as produced by the dense layer. A larger dimension of units was required for the 4/7 code rate as opposed to uncoded 8 bit message due to the coding gain required to match the Hamming(7,4) code.

Layer	Units Code Rate 7/4	Units Uncoded 8 Bit	Group
Input layer	2^K	2^K	Input
Dense layer	256	256	Skip block
Batch normalisation	-	-	
Swish activation	-	-	
Dense layer	128	16	
Batch normalisation	-	-	
Swish activation	-	-	
Dense layer	256	256	
Batch normalisation	-	-	
Swish activation	-	-	
Dense layer	$2N$	$2N$	$2N$ linear block
Linear activation	-	-	
Reshape $[N, 2]$ layer	-	-	
Dense layer	2	2	Output $[N, 2]$
Tanh activation	-	-	
Energy normalisation	-	-	

The training procedure is illustrated in Figure 4, which shows the three stages of the proposed disjoint training regime. This approach consists of training three models: a local transmitter model $Tx(x, \theta_t)$, a local channel/receiver model $Rx_L(z, \theta_l)$, and a remote receiver model $Rx_R(r, \theta_r)$, separated by a channel $h(z)$ which is not connected to the network models. The local channel/receiver model does not receive inputs r from the simulated channel; instead, it takes its inputs directly from the output of the transmitter model $z = Tx(x, \theta_t)$. During training a feedback channel is required, allowing the average value to be captured for the remote loss per batch along with remote estimates $p(y|r)$ for each item in the batch. Only one network is trained at each stage.

Table 2. The receiver network, consisting of three groups for input, feature learning (skip block), and output. The dimension of units are shown for each dense layer, with subsequent layers producing the same shape output as the preceding dense layer. A larger network was required to achieve the 4/7 code rate as opposed to the uncoded 8 bit message.

Layer	Units Code Rate 7/4	Units Uncoded 8 Bit	Group
Input layer	$[N, 2]$	$[N, 2]$	Input
Flatten layer	-	-	
Dense layer	256	256	Skip block
Batch normalisation	-	-	
Swish activation	-	-	
Dense layer	128	16	
Batch normalisation	-	-	
Swish activation	-	-	
Dense layer	256	256	
Batch normalisation	-	-	
Swish activation	-	-	
Dense layer	2^K	2^K	Output
Softmax activation	-	-	

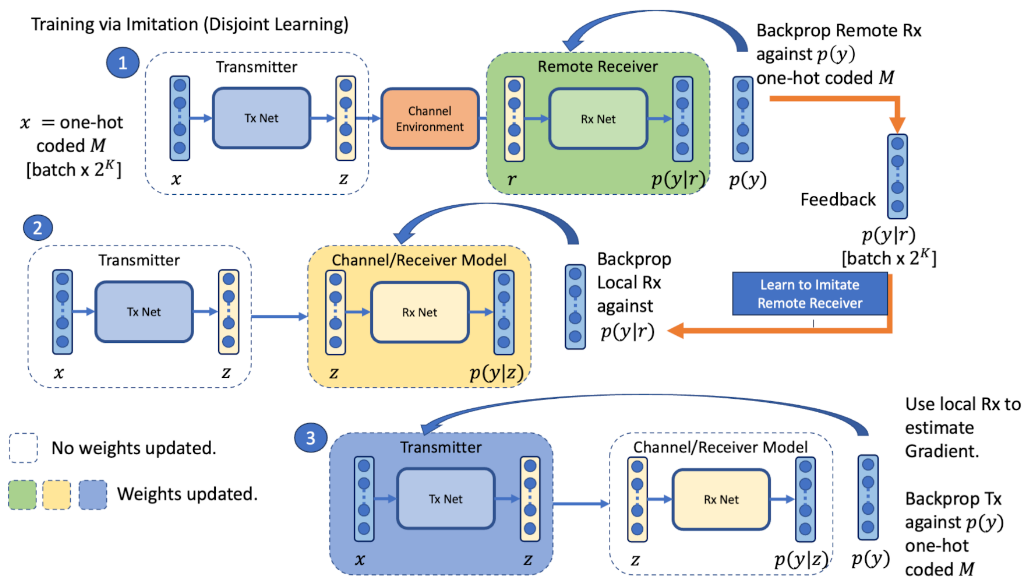


Figure 4. The three stages of the training procedure consist of a forward pass through the transmitter $z = Tx(x, \theta_t)$, channel $r = h(z)$, and remote receiver $p(y|r) = Rx(r, \theta_r)$. The remote receiver estimates $p(y|r)$ are obtained through the feedback channel. The second stage trains the local channel/receiver model using the KL divergence loss between the local channel/receiver estimates $p(y|z)$ and remote receiver estimates $p(y|r)$. The third stage trains the transmitter using the local receiver as a proxy to enable end-to-end backpropagation.

To generate the same sequences of random messages in each training iteration, both sides are initialised with the same random seed at the start of each batch. In the first stage, the local transmitter remains frozen and provides the forward pass for the batch of symbols $z = Tx(x, \theta_t)$ that are sent to the remote receiver over the simulated channel $r = h(z)$. The remote receiver is trained with SGD against the cross-entropy loss between the true and estimated message labels in Equation (1). The remote receiver probability estimates $p(y|r)$, along with the mean loss, are sent over the feedback channel to the local transmitter. In the second stage, the local receiver is trained to imitate the remote receiver using the Kullbach-Leibler (KL) divergence loss in Equation (4). The aim is to minimise the difference between the estimated probabilities at the remote receiver

$p(y|r) = Rx_R(r, \theta_r)$ given the channel symbols, and the estimated probabilities at the local channel/receiver $p(y|z) = Rx_L(z, \theta_l)$ given the local transmitter symbols z. This allows the local channel/receiver to learn how to act as a proxy for the remote receiver without an explicit model of the channel by imitating the estimation produced at the remote receiver. As indicated by Equation (5), the gradient for the connected local transmitter Tx and the local channel/receiver Rx_L can be described as a weight update for the combined weights $\theta_{t,l}$ of the end-to-end connected model $\theta_{t,l}$. Because the feedback channel provides $p(y|r)$ from the remote receiver, the backpropagation at the transmitter side is no longer dependent on an assumed channel function.

$$\mathcal{L}_{stage2}[p(y|r), p(y|z)] = \sum_{i=1}^{2^K} p(y_i|r) \log \frac{p(y_i|r)}{p(y_i|z)} \tag{4}$$

$$\nabla_{\theta_{t,l}} = \frac{\partial Tx(x, \theta_{t,l})}{\partial \theta_{t,l}} \frac{\partial Rx_L(Tx(x, \theta_{t,l}), \theta_{t,l})}{\partial Tx(x, \theta_{t,l})} \frac{\partial \mathcal{L}_{\text{stage2}}[p(y|r), Rx_L(Tx(x, \theta_{t,l}), \theta_{t,l})]}{\partial Rx_L(Tx(x, \theta_{t,l}), \theta_{t,l})} \tag{5}$$

Both Stage 1 and Stage 2 make use of a larger batch size than the third stage, which we set at 320 samples in stages 1 and 2 and 32 samples in Stage 3. In the third stage, a forward pass through the transmitter is made for a new batch. The local channel/receiver is used to calculate the cross-entropy loss against the true messages $\mathcal{L}[p(y), p(y|z)]$. The local channel/receiver estimates are conditioned on the output of the local transmitter z, rather the output of the simulated channel r as is the case on the remote receiver. Updates resulting from the backpropagation process occur only on the local transmitter, as the local channel/receiver weights are frozen during this step. The label noise introduced in the second stage enables the transmitter to learn appropriate IQ symbols to assist the remote receiver in the labelling task. To demonstrate that this approach has an effect, we performed training with no feedback, in which the second stage of the algorithm updates a local receiver model against the true message using the cross-entropy loss instead of optimising toward the remote distribution. In Section 4, we demonstrate that training the local channel/receiver to imitate the remote receiver produces an observable difference in performance in comparison to the same algorithm without feedback.

Energy normalisation is applied to constrain the output of the transmitter such that $||x||_2^2 \leq 1$, as defined in Equation (6), where the learned code $x(t)$ with length L is divided by its scaled Euclidean norm to produce the transmit symbols $z(t)$.

$$z(t) = \frac{x(t)}{\sqrt{\sum_{i=1}^{L} x(i)^2 / L}} \tag{6}$$

Each of the networks are trained using the Adam algorithm [30], and we combine stochastic weight averaging (SWA) [35] with a cyclical learning rate schedule [36] which oscillates between learning rates of 0.0001 and 0.001. In this work, we simulate the channel transfer function as described in Section 3.3. This allows the signal to noise ratio (SNR) dB to be randomised during training of the remote receiver; however, in an over-the-air setting, the SNR dB parameter cannot be set explicitly.

3.3. Simulated Channel Functions

Comparisons between models were made in simulated channel environments for AWGN, Rayleigh, and Rician fading as well as for an AWGN channel with nonlinear amplifier effects, namely, power amplifier Additive White Gaussian Noise (PA-AWGN). In the AWGN channel, the transfer function adds a noise term $n(t)$ to the symbols output by the transmitter in Equation (7).

$$r(t) = z(t) + n(t) \tag{7}$$

In the Rayleigh fading channel, a series of complex fading coefficients $a(t) = \frac{1}{\sqrt{2}}|a|$ are sampled from the complex standard normal distribution $a \sim \mathcal{CN}(0,1)$. These coefficients are applied to scale the transmitter symbols before adding the noise term in Equation (8).

$$r(t) = a(t)z(t) + n(t) \tag{8}$$

The Rician fading channel has the same structure as Equation (8), except that the fading coefficients are drawn from a parameterised complex normal distribution. The mean $\mu = \sqrt{K/(2(K+1))}$ and standard deviation $\sigma = \sqrt{1/(2(K+1))}$ are both determined by the Rician factor K, which in our simulations we define as $K = 10$. The coefficients are then drawn from the complex normal distribution $a \sim \mathcal{CN}(\mu, \sigma^2)$ and applied to scale the transmitter symbols.

The PA-AWGN assumes a Rapp model of a solid state high power amplifier (SSPA) [37] that is applied to the output of the transmitter in Equation (9). The parameters for the model include the limiting output amplitude A_0, a gain parameter ν, and a smoothness parameter p; in our simulations, we configure $A_0 = 1$, $\nu = 1$, and $p = 5$. The nonlinearity operates on the magnitude of the transmitter output, and is multiplied by the complex exponent of the argument of the transmitter output $A = |z(t)|$. In the PA-AWGN channel, AWGN is applied after amplification.

$$\begin{aligned} g(A) &= \nu \frac{A}{\left(1 + \left[\left(\frac{\nu A}{A_0}\right)^2\right]^p\right)^{1/2p}} \\ z'(t) &= g(|z(t)|)e^{j\angle z(t)} \end{aligned} \tag{9}$$

The noise term $n(t)$ in each of the channel models above is drawn from the complex normal distribution. When simulating the channel function, we define the desired level of SNR or ratio of energy per bit to the noise E_b/N_0 provided in dB. As the models learn a coded modulation with a code rate K/N, we convert this quantity to the ratio of energy per symbol to noise $E_s/N_0 \text{ dB} = E_b/N_0 \text{ dB} + 10log_{10}(K/N)$ and use the linear ratio $E_s/N_0 = 10^{E_s/N_0 \text{ dB}/10}$ to separate terms for E_s and N_0. The term E_s is estimated directly from the L transmitter IQ symbols $E_s = \sum_{t=1}^{L} z(t)^2/L$ and $N_0 = E_s/(E_s/N_0)$. The noise is then sampled from the complex normal distribution $n(t) \sim \mathcal{C}N(0, \sigma^2)$ using the variance $\sigma^2 = N_0/2$.

4. Results and Discussion

In this section, we evaluate the proposed method in the AWGN, Rician and Rayleigh fading, and PA-AWGN channels. In the AWGN channel, we train and compare the joint model and the proposed disjoint model for the 8 bit uncoded and Hamming(7,4) code rates. We additionally draw comparisons between receiver tuning for the joint model and the disjoint model. Receiver tuning is performed by training the joint model in the Rician fading channel and tuning the receiver in the Rayleigh fading channel. We make comparisons with receiver tuning by training the joint model on the AWGN channel and tuning the receiver in the PA-AWGN channel. This is performed for both code rates. We present a comparison between the proposed disjoint training method requiring feedback against the training without feedback. These results are reported in the Rayleigh fading channel. In addition, we present results for quantisation of the feedback, which can reduce the overall channel usage required during training.

The joint and disjoint learning methods for the 8 bit message are compared with uncoded binary phase shift keying (BPSK) under several channels in Figure 5. The proposed disjoint learning process provides slightly better performance than the joint learning procedure under AWGN (Figure 5a). In the Rician fading channel, disjoint learning achieves lower block error rate (BLER) than the joint learning method (Figure 5b), whereas disjoint and joint learning produce similar BLER in the Rayleigh fading channel (Figure 5c). Receiver tuning leverages the joint dense network from the Rician fading channel and updates the receiver under the Rayleigh fading channel (Figure 5c). Receiver tuning does not reach the same level of BLER as the other methods.

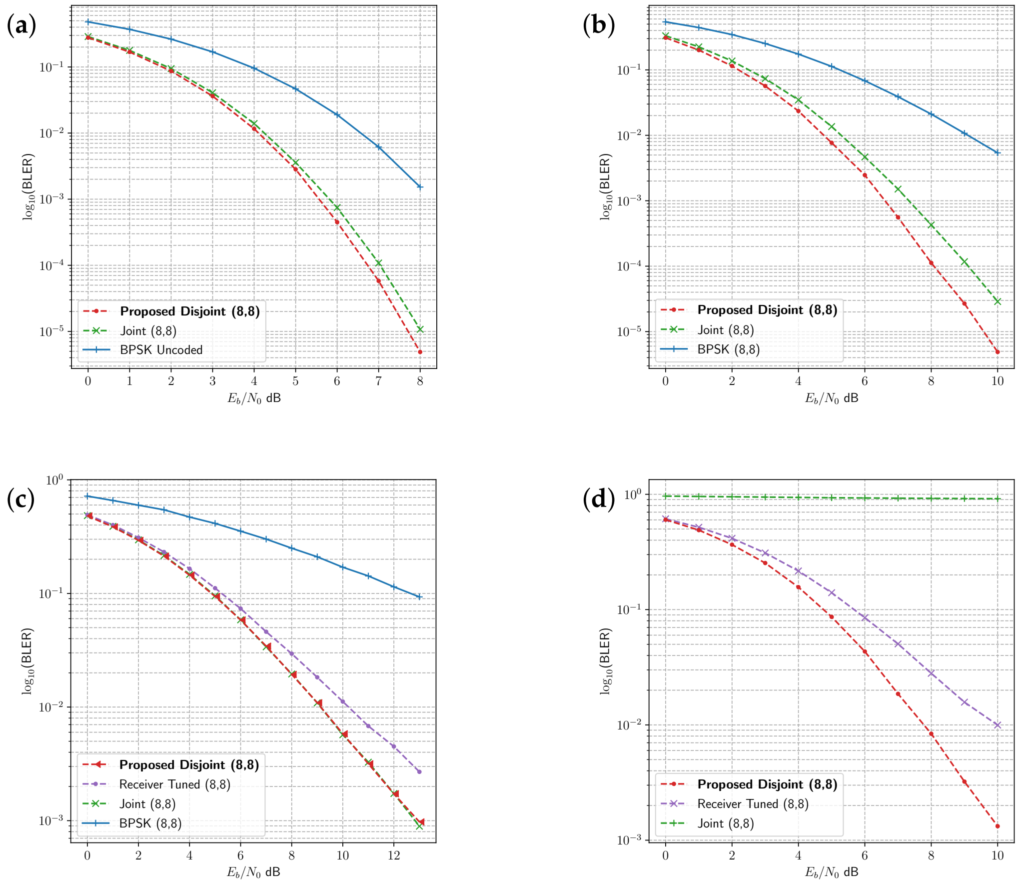


Figure 5. Comparison of BLER in the four channel environments for the uncoded 8 bit message. The joint learning, disjoint learning and BPSK modulation are compared in the (**a**) AWGN channel, (**b**) Rician fading channel, and (**c**) Rayleigh fading-channel. (**d**) Comparison between joint learning, disjoint learning, and receiver tuning in AWGN with PA-AWGN non-linearity.

Joint and disjoint learning methods are compared to the Hamming(7,4) code in Figure 6 in the AWGN (Figure 6a), Rician (Figure 6b), and Rayleigh (Figure 6c) fading channels. Both the joint and disjoint methods exhibit very similar or slightly better performance as maximum likelihood decoding (MLD) for the Hamming(7,4) code in each of these channels. Receiver tuning is repeated for the (7,4) code in Figure 6c, adapting the joint model receiver trained under Rician fading to the Rayleigh fading channel. While the performance is close to the other codes, it does not achieve the same BLER as the disjoint method with the transmitter optimised for the channel environment.

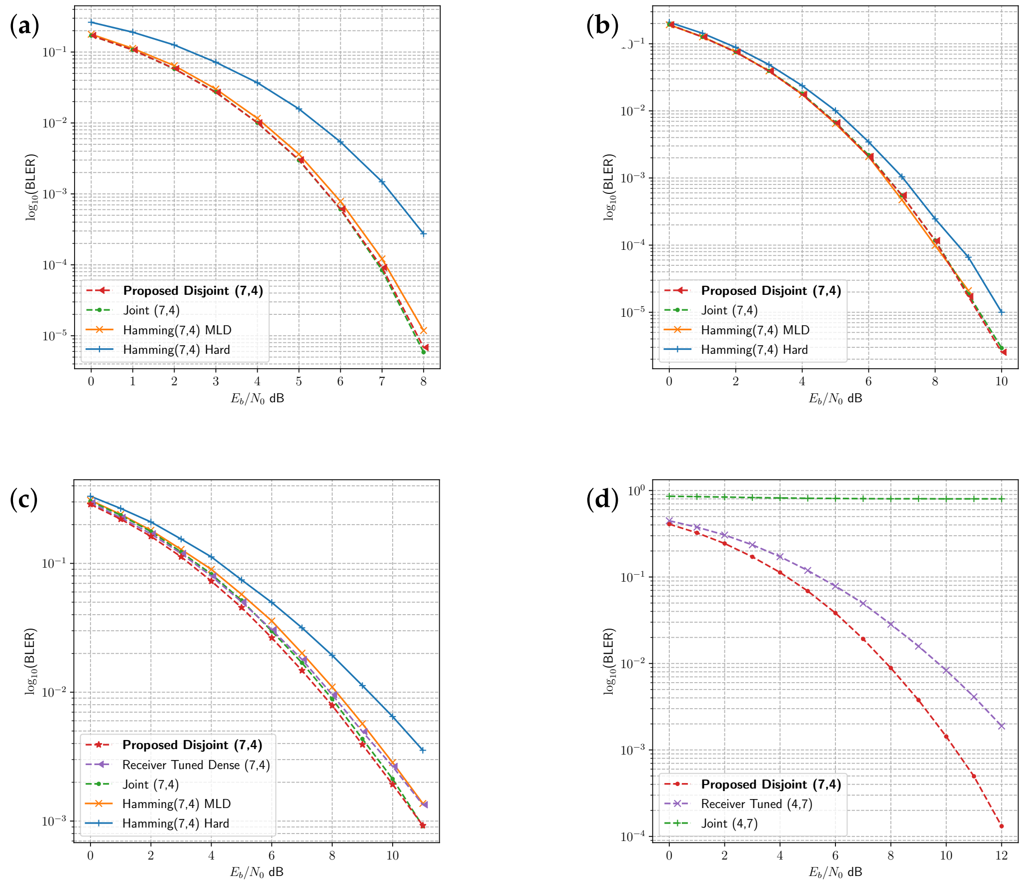


Figure 6. Comparison of BLER in the four channel environments for the $K = 4$, $N = 7$ code rate. The joint learning, disjoint learning and the BPSK modulated Hamming(7,4) code are compared in the (**a**) AWGN channel, (**b**) Rician fading channel, and (**c**) Rayleigh fading-channel. (**d**) Comparison between joint learning, disjoint learning, and receiver tuning in AWGN with PA-AWGN non-linearity.

There is a difference in architecture between the joint and proposed disjoint models for the transmitter and receiver described in Section 3.2. The combination of the residual connections and additional dense layers increases the size of the disjoint models slightly, and contribute to the gain over the joint model. In comparison to uncoded BPSK modulation, the joint and proposed models learn a continuous code that is non-zero in both IQ coordinates; the resulting code is more complex than BPSK modulation, which is non-zero on the in-phase (I) axis. The performance of a code is related to the minimum squared distance between all codes [38]. Ideally, the transmitter should learn a code that has a large minimum Euclidean distance. Taking for example the $K = 4, N = 7$ learned code, we can compute the minimum ($d_{E_{min}}$), mean ($E[d_E]$), and variance ($Var[d_E]$) of the Euclidean distances for each of the proposed $K = 4, N = 7$ disjoint models, as shown in Table 3. The reference Hamming(7,4) code with a minimum binary distance (d_{min}) of 3 is included for comparison. The disjoint model has learned a slightly different code under each of the channels, each with a slightly different value for $d_{E_{min}}$. While $d_{E_{min}}$ is not always larger than the computed value for the Hamming(7,4) code, $E[d_E]$ is slightly larger, and the $Var[d_E]$ is quite low in comparison. We would expect that the learned code would perform slightly better in those channels where $d_{E_{min}}$ is larger than the reference code, which is indeed the case for AWGN. While the learned code in the Rayleigh channel has a slightly lower minimum Euclidean distance, it appears that the $E[d_E]$ and low $Var[d_E]$ may contribute to the overall performance of the learned code.

Table 3. Computed minimum ($d_{E_{min}}$), mean ($E[d_E]$), and variance ($Var[d_E]$) of the Euclidean distances between 2^K messages. Distance measures are shown for the reference Hamming(4,7) and the disjoint $K = 4, N = 7$ code trained in each channel environment. The minimum binary distance (d_{min}) is provided for the Hamming code, but is not applicable for the learned continuous codes.

Code	d_{min}	$d_{E_{min}}$	$E[d_E]$	$Var[d_E]$
Hamming(7,4)	3	3.46	3.83	0.21
Disjoint AWGN (7,4)	-	3.51	3.85	0.08
Disjoint Rician (7,4)	-	3.44	3.85	0.07
Disjoint Rayleigh (7,4)	-	3.37	3.86	0.05
Disjoint PA-AWGN (7,4)	-	3.06	3.85	0.08

To further investigate the effect of the channel on tuning and disjoint learning, we compared the joint model trained on AWGN with a receiver tuned model and the disjoint model under the PA-AWGN channel. Figures 5d and 6d show the BLER for the uncoded 8 bit message and the 4/7 code rate, respectively. In both cases, the AWGN joint model is unable to provide decoding for learned symbols under the PA-AWGN channel. However, the receiver tuned model derived from the same joint AWGN model learns to optimise the receiver, allowing it to classify messages in this environment. The advantage of the proposed disjoint learning algorithm is indicated by the improvement in performance over the receiver tuned model due to training both the transmitter and receiver.

Because the proposed disjoint model outperforms the receiver tuned model, it is clear that the transmitter model is learning a code that is specifically optimised to the target channel environment where it is trained. This is evident in the distance measurements of the $K = 4, N = 7$ code presented in Table 3. To evaluate the difference between the learned codes, we computed the BLER performance for models which were not trained on two of the selected channel environments. Figure 7a presents the BLER for the disjoint models which were not trained on the Rayleigh fading channel in comparison with the optimal disjoint model for that channel. The performance of the disjoint models optimised for the AWGN and Rician fading channel are similar to, but do not exactly match, the same performance of the optimised Rayleigh fading model. These two models have been optimised for slightly simpler channels than the Rayleigh fading channel. The Rician fading channel has slightly different fading characteristics from the Rayleigh fading channel, and the Rician model is closer in performance. The AWGN channel has no fading effects, and the resulting model has higher BLER than both of the other fading models. However, there is a large difference between the performance of the disjoint model optimised for the PA-AWGN channel and the other models. The performance is reversed in Figure 7b, where the PA-AWGN model is the optimal model. By imitating the remote receiver, the local channel/receiver enables the transmitter to learn codes which are optimised for the channel environment and which can be applied in channels with similar characteristics. However, it is possible for channel environments to differ significantly, as illustrated in Figure 7. The nonlinear effects of the amplifier are unique to the PA-AWGN channel, and are not shared with the other channels. In a practical wireless communications system, it is necessary to detect when the channel changes significantly (i.e., when performance degrades) and to either adapt using OAL and/or develop DL methods for adaptive modulation and coding schemes [38] that can select from multiple learned codes.

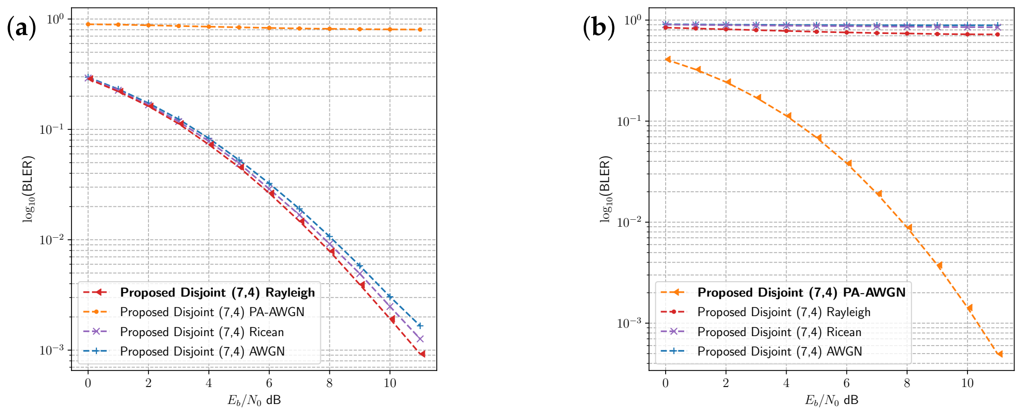


Figure 7. Comparison of BLER performance for the proposed disjoint models without retraining on targeted channel environments. Indicating the transmitter learns a code that is optimised for the targeted channel. (**a**) Comparison of BLER performance without retraining in the Rayleigh fading channel. (**b**) Comparison of BLER performance without retraining in the PA-AWGN channel.

The proposed method enables the transmitter to learn codes that are optimised during training for the observed channel environment. However, the question arises as to what extent imitating the remote receiver is helpful in achieving optimisation at the transmitter. Is it possible to achieve the same optimisation by simply training the local receiver against the true target message? We compared this no-feedback approach against the disjoint Learning method in Figure 8, where disjoint learning with feedback strongly outperforms learning without feedback. It is not sufficient to train the local receiver against a noiseless channel; instead, by imitating the remote receiver, enough information about the channel distortion is provided to the transmitter model during backpropagation to enable it to learn optimal symbols for the current channel condition. This is clearly indicated in both Figures 5 and 6, where the disjoint method either outperforms or matches the joint learning method, achieving optimal BLER (in the case of the Hamming(7,4) code).

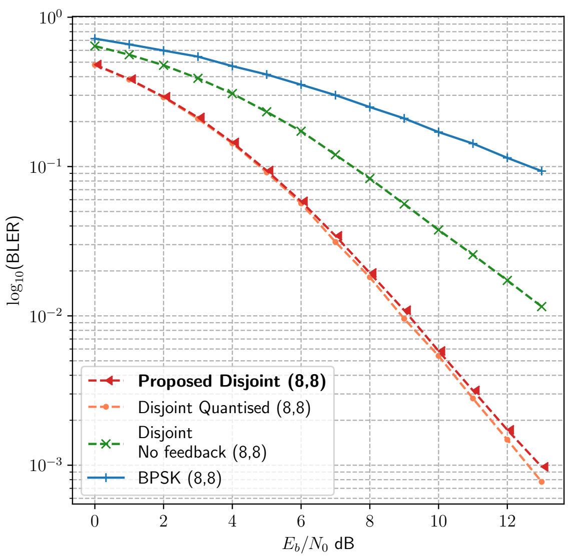


Figure 8. Comparison between learning without feedback, learning with soft values, and learning with quantised values in the Rayleigh fading channel.

Feedback of soft values during disjoint training does require a large amount of data, depending on the message size; for example, in an uncoded 8 bit message, the feedback stage requires a batch size of 320×256 soft values. It is desirable to reduce the amount of information that needs to be sent over the feedback channel during learning. One possible method is to simply take the $\arg\max p(y|z)$ output at the remote receiver and feed back the integer indices for learning at the local channel/receiver; this reduces the amount of data to the batch size (320×1). As these integer values can be translated to a one-hot encoding on the transmitter side, the local channel/receiver then learns to imitate the remote receiver through the cross-entropy loss. Figure 8 compares the performance resulting from training with reduced information (Disjoint Quantised) as opposed to soft values (Proposed Disjoint), and indicates no loss of performance under the Rayleigh fading channel.

Our results show that the learning process in the transmitter is dependent on the local channel/receiver model. This is indicated by the ability to learn an equivalent or better performing code than the joint AE as well as by the difference in performance in different channels. The feedback of the estimates $p(y|r)$ from the remote receiver contains implicit information about the channel environment. This implicit information is conveyed by the errors made at the remote receiver, which can be regarded as a kind of classification label noise, such as the type of regularisation introduced in [28]. Hence, by learning to imitate the remote receiver, the local channel/receiver learns to make the same errors over the course of learning. Unlike traditional supervised learning for classification, in which a model is optimised against a static set of target labels, the proposed learning process gradually changes all three models (the local transmitter, local channel/receiver, and remote receiver). The implication is that all three models are jointly optimised. In order to improve performance at the remote receiver, the transmitter alters the learned code based on the distance between the local channel/receiver estimate $p(y|z)$ and the remote receiver estimate $p(y|r)$. The need for backpropagation over an unknown channel is mitigated, as the information required to learn an optimal code is contained in the feedback of the estimates for $p(y|r)$ from the remote receiver.

While we have demonstrated equivalent or better performance compared to the joint model, our work has a number of limitations. First, we assumed perfect synchronisation and did not apply matched filtering or any timing, phase, or frequency distortions. Second, for the purposes of discussion, we have limited our study to the domain of short codes. Third, the method requires high use of a feedback channel, similar to the RL-based methods. However, we have shown that it is possible to reduce the feedback channel usage; instead of learning to approximate the soft values for $p(y|r)$ estimated at the remote receiver, it is possible to train against the $\arg\max p(y|r)$ without loss of performance. Finally, our method does not explicitly model the channel in the way that a GAN provides a separate channel model which can be reused outside of the training process. Instead, the local channel/receiver provides an implicit distortion to the transmitter in order to enable optimisation. Our approach represents a simplification over other training methods, requiring fewer models than the GAN approach by omitting the generator and discriminator models. The proposed method is able to take advantage of backpropagation directly, as opposed to the gradient approximation applied in RL methods, and does not require a complex coordinating protocol such as the one used in cooperative multi-agent learning.

5. Conclusions and Further Research Work

To date, disjoint learning methods have focused on simulation of the channel via GAN or gradient approximation through finite difference methods or RL approaches. In this paper, we have presented an additional approach to disjoint learning by learning to imitate the remote receiver. We have demonstrated equivalent performance to joint learning in AWGN, Rician, and Rayleigh fading channels, and shown that learning to imitate has the advantage of optimising both the transmitter and receiver, as opposed to receiver tuning, which can only adapt the remote receiver after joint learning. By comparing the distance metrics for learned codes, the performance of models in different channels, and the difference in performance between training with and without feedback, we have provided evidence that our proposed method is able to optimise the local transmitter and remote receiver models without an assumed channel model. The local channel/receiver model has no explicit knowledge of the channel, and by imitating the remote receiver, provides enough implicit channel information to enable the transmitter to learn optimal codes for the channel environment.

The limitations described in Section 4 provide an opportunity for future investigation. The assumption of perfect synchronisation can be addressed by incorporating additional channel perturbations and matched filtering. Further investigation into longer codes can be facilitated by incorporating bitwise estimation and concatenated codes. Bitwise estimation differs from symbolwise classification, and may alter the optimisation of the local channel/receiver during training. This method can find potential applications in joint-source coding and semantic coding, which have both benefited from the use of AE. In order to address changing channel conditions, it is possible to investigate alterations to the architecture to support adaptive modulation and coding schemes, or alternately to determine how to appropriately retrain the transmitter and receiver models under changing channel environments. In addition, future scope remains for investigating real-time training requirements over physical hardware. Even though this method makes no assumptions about the channel model, it is possible to train the models offline and adapt both transmitter and receiver models in a deployed scenario as opposed to receiver only tuning.

As an alternative to channel approximation and gradient approximation methods in disjoint learning, learning to imitate the remote receiver provides an additional method of disjoint learning without an assumed channel, and offers a simplified training procedure that can be applied to over-the-air learning in wireless communication systems.

Author Contributions: C.P.D.: conceptualization, data curation, formal analysis, investigation, methodology, project administration, resources, software, validation, visualization, writing—original draft, writing—review and editing. I.S.: conceptualization, resources, project administration, supervision, writing—review and editing. R.C.D.: resources, project administration, supervision, writing—review and editing. S.S.-S.: writing—review and editing, resources. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by UniSQ-DSTG Postgraduate Research Scholarship 2021–2024 on the 'Design of Efficient Artificial Intelligence Algorithms for Future Communication Systems'. It is funded by the Department of Defence, Commonwealth of Australia under DSP Scholarship (Project-Based) Agreement 10254.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. O'Shea, T.; Hoydis, J. An Introduction to Deep Learning for the Physical Layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [CrossRef]
2. Li, P.; Pei, Y.; Li, J. A Comprehensive Survey on Design and Application of Autoencoder in Deep Learning. *Appl. Soft Comput.* **2023**, *138*, 110176. [CrossRef]
3. Dörner, S.; Cammerer, S.; Hoydis, J.; Brink, S.T. Deep Learning Based Communication Over the Air. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 132–143. [CrossRef]
4. Stark, M.; Aoudia, F.A.; Hoydis, J. Joint Learning of Geometric and Probabilistic Constellation Shaping. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]
5. Proakis, J.G.; Salehi, M. *Digital Communications*, 5th ed.; McGraw-Hill: New York, NY, USA, 2008; pp. 850–851, 1055–1058, ISBN 978-007-126378-8.
6. Ye, H.; Li, G.Y.; Juang, B.H. Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 114–117. [CrossRef]

7. Jiang, W.; Schotten, H.D. Deep Learning for Fading Channel Prediction. *IEEE Open J. Commun. Soc.* **2020**, *1*, 320–332. [CrossRef]
8. Yang, Y.; Li, Y.; Zhang, W.; Qin, F.; Zhu, P.; Wang, C.X. Generative-Adversarial-Network-Based Wireless Channel Modeling: Challenges and Opportunities. *IEEE Commun. Mag.* **2019**, *57*, 22–27. [CrossRef]
9. O'Shea, T.J.; Roy, T.; West, N.; Hilburn, B.C. Physical Layer Communications System Design Over-the-Air Using Adversarial Networks. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 529–532. [CrossRef]
10. O'Shea, T.J.; Roy, T.; West, N. Approximating the Void: Learning Stochastic Channel Models from Observation with Variational Generative Adversarial Networks. In Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019; pp. 681–686. [CrossRef]
11. Ye, H.; Li, G.Y.; Juang, B.F.; Sivanesan, K. Channel Agnostic End-to-End Learning Based Communication Systems with Conditional GAN. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–5. [CrossRef]
12. Ye, H.; Liang, L.; Li, G.Y.; Juang, B.H. Deep Learning-Based End-to-End Wireless Communication Systems With Conditional GANs as Unknown Channels. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3133–3143. [CrossRef]
13. Raj, V.; Kalyani, S. Backpropagating Through the Air: Deep Learning at Physical Layer Without Channel Models. *IEEE Commun. Lett.* **2018**, *22*, 2278–2281. [CrossRef]
14. Aoudia, F.A.; Hoydis, J. End-to-End Learning of Communications Systems Without a Channel Model. In Proceedings of the 2018 52nd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 28–31 October 2018; pp. 298–303. [CrossRef]
15. Aoudia, F.A.; Hoydis, J. Model-free training of end-to-end communication systems. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2503–2516. [CrossRef]
16. Cammerer, S.; Aoudia, F.A.; Dörner, S.; Stark, M.; Hoydis, J.; Brink, S.t. Trainable Communication Systems: Concepts and Prototype. *IEEE Trans. Commun.* **2020**, *68*, 5489–5503. [CrossRef]
17. Sahai, A.; Sanz, J.; Subramanian, V.; Tran, C.; Vodrahalli, K. Learning to Communicate with Limited Co-design. In Proceedings of the 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 24–27 September 2019; pp. 184–191. [CrossRef]
18. Goodfellow, I. Nips 2016 tutorial: Generative adversarial networks. *arXiv* **2016**, arXiv:1701.00160.
19. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
20. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
21. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
22. Dörner, S.; Henninger, M.; Cammerer, S.; Brink, S.t. WGAN-based Autoencoder Training Over-the-air. In Proceedings of the 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Online, 26–29 May 2020; pp. 1–5. [CrossRef]
23. Williams, R.J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.* **1992**, *8*, 229. [CrossRef]
24. Sutton, R.S.; McAllester, D.; Singh, S.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Adv. Neural Inf. Process. Syst.* **1999**, *12*.
25. Zhang, B.; Van Huynh, N. Deep Deterministic Policy Gradient for End-to-End Communication Systems without Prior Channel Knowledge. *arXiv* **2023**, arXiv:2305.07448.
26. de Vrieze, C.; Barratt, S.; Tsai, D.; Sahai, A. Cooperative multi-agent reinforcement learning for low-level wireless communication. *arXiv* **2018**, arXiv:1801.04541.
27. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* **2021**, *64*, 107–115. [CrossRef]
28. Xie, L.; Wang, J.; Wei, Z.; Wang, M.; Tian, Q. Disturblabel: Regularizing cnn on the loss layer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4753–4762. [CrossRef]
29. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
32. Veit, A.; Wilber, M.J.; Belongie, S. Residual networks behave like ensembles of relatively shallow networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 550–558.
33. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015*; Bach, F., Blei, D., Eds.; Proceedings of Machine Learning Research PMLR; JLMR.org: Lille, France, 2015; Volume 37, pp. 448–456.
34. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.

35. Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; Wilson, A.G. Averaging weights leads to wider optima and better generalization. *arXiv* **2018**, arXiv:1803.05407.
36. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 464–472. [CrossRef]
37. Rapp, C. Effects of HPA-nonlinearity on a 4-DPSK/OFDM-signal for a digital sound broadcasting signal. *ESA Spec. Publ.* **1991**, *332*, 179–184.
38. Goldsmith, A.J.; Chua, S.G. Adaptive coded modulation for fading channels. *IEEE Trans. Commun.* **1998**, *46*, 595–602. [CrossRef]

6.3 Links and Implications

The article contributes a simplification of the training process for OAL by demonstrating that it is not necessary to perform Gradient approximation or use Channel approximation to model the channel explicitly. Instead, it is possible to train a local receiver against the remote receiver estimates on the transmitter side, which can be leveraged to train the transmitter in an E2E manner. This method is demonstrated to learn implicit knowledge of the channel through imitating the noisy errors made by the remote receiver. In addition, the article shows that it is possible to reduce the feedback channel usage by providing only the quantised estimates from the remote receiver without negatively impacting performance of the trained system. The method is also shown to jointly optimise the transmitter and receiver by comparing resulting performance against receiver tuning with a pre-trained model.

CHAPTER 7: PAPER 4 - DEEP LEARNING BASED OVER-THE-AIR TRAINING OF WIRELESS COMMUNICATION SYSTEMS WITHOUT FEEDBACK

7.1 Introduction

This chapter presents a copy of the article published in Sensors (vol. 24 (2024), no. 10, p2993, ISSN: 1424-8220, doi:10.3390/s24102993. https://www.mdpi.com/1424-8220/24/10/2993).

In OAL, Channel approximation develops a separate DL model of the true channel environment and is applied in training transmitter and receiver E2E. A series of methods requiring feedback during iterative training have been proposed in the literature, however, the use of a feedback channel increases channel usage during training and potentially exposes the training procedure to adversarial attacks, such as replay attacks. Therefore, it is desirable to train without the use of a feedback channel. Prior work has also suggested the use of a pre-trained transmitter to generate batches for the training of a channel model on the receiver side. However, an information carrying signal remains vulnerable in an adversarial environment. Hence, this article trains a channel model on random samples over the true channel environment and develops a transmitter and receiver E2E on the resulting channel model. The proposed method is shown to perform equivalently to the canonical AE in a number of channel environments. In addition, the channel model is developed from the MDN modelling approach, which is a simpler approach with less training overhead than either GAN or diffusion networks that have been proposed in the literature.

Research Highlights

- An iterative OAL algorithm is proposed for the development of transmitter, receiver and channel model which does not require continuous feedback between transmitter and receiver.

- The article describes application of the MDN for the approximation of the channel transfer function, and demonstrates approximation for several simulated channels including the AWGN, Rician fading, Rayleigh fading and power amplifier AWGN channels.
- Intermittent measurement of BLER for transmitter and receiver models via the generative channel is shown to highly correlated with the BLER derived from the true channel and has suitable application as training stopping criteria and for monitoring of learning process.
- Finally, the article demonstrates that the performance for the resulting transmitter and receiver models are equivalent to or better than the end-to-end model which is trained with an assumed channel model. This is shown for AWGN, Rician fading, Rayleigh fading and non-linear power amplifier distortions over AWGN simulated channels.

7.2 Published Article 4

sensors

Article

Deep Learning Based Over-the-Air Training of Wireless Communication Systems without Feedback

Christopher P. Davey [1,*], Ismail Shakeel [2], Ravinesh C. Deo [1,*] and Sancho Salcedo-Sanz [3]

1 School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, QLD 4300, Australia
2 Spectrum Warfare Branch, Information Sciences Division, Defence Science and Technology Group (DSTG), Edinburgh, SA 5111, Australia; ismail.shakeel@defence.gov.au
3 Department of Signal Processing and Communications, Universidad de Alcalá, 28805 Alcalá de Henares, Madrid, Spain; sancho.salcedo@uah.es
* Correspondence: christopher.davey@unisq.edu.au (C.P.D.); ravinesh.deo@unisq.edu.au (R.C.D.)

Abstract: In trainable wireless communications systems, the use of deep learning for over-the-air training aims to address the discontinuity in backpropagation learning caused by the channel environment. The primary methods supporting this learning procedure either directly approximate the backpropagation gradients using techniques derived from reinforcement learning, or explicitly model the channel environment by training a generative channel model. In both cases, over-the-air training of transmitter and receiver requires a feedback channel to sound the channel environment and obtain measurements of the learning objective. The use of continuous feedback not only demands extra system resources but also makes the training process more susceptible to adversarial attacks. Conversely, opting for a feedback-free approach to train the models over the forward link, exclusively on the receiver side, could pose challenges to reliably end the training process without intermittent testing over the actual channel environment. In this article, we propose a novel method for the over-the-air training of wireless communication systems that does not require a feedback channel to train the transmitter and receiver. Random samples are transmitted through the channel environment to train a mixture density network to approximate the channel distribution on the receiver side of the network. The transmitter and receiver models are trained with the resulting channel model, and the transmitter can be deployed after training. We show that the block error rate measurements obtained with the simulated channel are suitable for monitoring as a stopping criterion during the training process. The resulting method is demonstrated to have equivalent performance to the end-to-end autoencoder training on small message sequences.

Keywords: deep learning; feedback-free training; trainable wireless communications systems; over-the-air training; neural networks

Citation: Davey, C.P.; Shakeel, I.; Deo, R.C.; Salcedo-Sanz, S. Deep Learning Based Over-the-Air Training of Wireless Communication Systems without Feedback. *Sensors* **2024**, *24*, 2993. https://doi.org/10.3390/s24102993

Academic Editors: Jose F. Monserrat and Changchuan Yin

Received: 14 March 2024
Revised: 19 April 2024
Accepted: 6 May 2024
Published: 8 May 2024

1. Introduction

Messages in a wireless communication system are sent from a transmitter over the air, via a channel environment, to a receiver whose aim is to recover the original message. A simplified depiction of such a wireless communications system is shown in Figure 1. The channel environment is significant in this type of communications system, as it distorts the message with perturbations such as noise and fading effects. These channel effects, along with imperfections within the electronics of both the transmitter and the receiver, present a challenge to the recovery of the original message. To improve accuracy at the receiver, the transmitter can code message bits to enable error correction at the receiver. It is also responsible for modulating the message bits and converting the modulation to a radio frequency (RF) signal suitable for sending over the wireless channel. At the receiver, the distorted RF signal must be detected, demodulated, and decoded in order to recover the original sequence of bits. Each of these steps is conventionally defined as separate signal processing blocks that are optimised independently of one another [1].

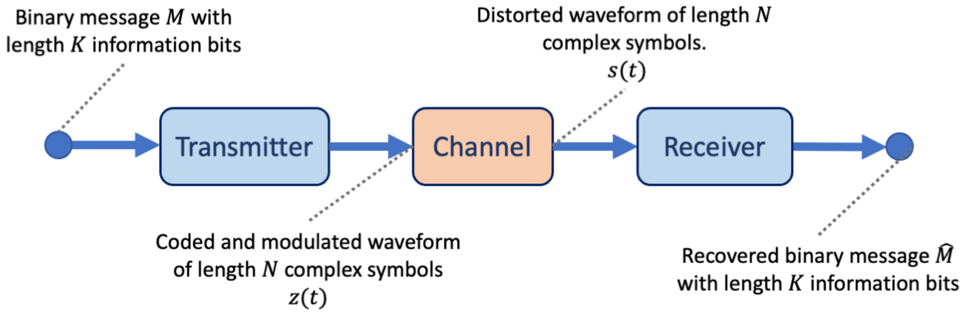


Figure 1. A simplified view of a wireless communications system. The transmitter takes in a K bit binary message M and codes and modulates the message producing transmitter symbols $z(t)$. These symbols are transmitted over a channel that produces noise and outputs symbols $s(t)$. The receiver is responsible for correcting the channel distortions and producing an estimate for the original message $\hat{M}$.

End-to-end deep learning (DL) for wireless communications systems has been proposed as an alternative design approach to that of block-based traditional design [1]. The primary advantage of DL over block-based design is the potential to perform end-to-end optimisation over observations of a complex channel environment. Especially where the channel environment may be too complex to be expressed mathematically. However, in [1], the channel environment is assumed and described by a differentiable channel transfer function that does not necessarily capture the description of a true channel environment. Instead of assuming a channel function, it is preferable to jointly optimise the DL-based transmitter and receiver over examples produced by the true channel. However, in DL this poses a genuine challenge. The backpropagation algorithm, which modifies the parameters of the model, cannot occur between the transmitter and receiver. This is because the calculation of the gradient for the model parameters cannot be determined without a differentiable channel function.

To overcome this limitation, over-the-air learning (OAL) methods have either applied gradient approximation [2,3] or trained a separate generative channel model to enable end-to-end backpropagation [4,5]. The primary limitations of gradient approximation are the requirement to sample several perturbations through the channel at each training iteration and continuous feedback of the receiver error. Continuous feedback increases channel usage and exposes the training process to eavesdropping and data poisoning attack by adversary communications systems [6]. In the generative channel modelling approach, the generative adversarial network (GAN) has been widely adopted to approximate the wireless channel distribution [4,5,7]. However, GAN training requires two models to learn a channel approximation, a generator and a discriminator model, and proceeds in two stages. First, by training the discriminator to recognise true channel symbols versus those produced by the generator model, and second by training the generator to fool the discriminator. This adversarial training regime adds complexity to the overall training process for the transmitter and receiver.

In our prior work, we proposed a disjoint OAL algorithm that trains the transmitter with a local receiver by imitating the errors made at the remote receiver [8]. The local receiver relied on a feedback channel to supply the remote error information. However, as in gradient approximation, the feedback channel increases channel use and is vulnerable to eavesdropping and data poisoning attack during training.

Reliance on continuous feedback is a vulnerability for the overall security of OAL of DL-based wireless communications systems. To realise OAL on energy-constrained devices such as in the internet of things (IoT), it is important to avoid complex training procedures which require training multiple models. Both of these considerations motivate the work in this article with the following aims:

- To simplify the training procedure for OAL learning of transmitter and receiver, by proposing an alternative to gradient approximation and eliminating the requirement for the use of a feedback channel, as well as by developing a simple channel model that does not require adversarial training against a discriminator, while still learning an accurate approximation of the observed channel distribution.
- To reduce the vulnerability of OAL training to eavesdropping and adversarial attacks by removing the use of a feedback channel and preventing transmission of information-carrying symbols over the true channel environment that could be intercepted and altered during training.

Motivated by these challenges, in this article, we investigate a method for OAL in wireless communication systems that can be performed on the receiver side. The proposed approach does not require continuous feedback and requires training only a single model to approximate the distribution of the true channel. Additionally, we determine that intermittent evaluation of the transmitter and receiver using the resulting channel model is a suitable method for determining training stopping criteria and provides a measurement appropriate for monitoring of the learning process.

The key contributions of this article are:

- We propose an iterative OAL algorithm for the development of a transmitter, receiver, and channel model § that does not require continuous feedback between transmitter and receiver.
- We discuss the application of the mixture density network (MDN) for the approximation of the channel transfer function. We also show the demonstration of approximation for several simulated channels, including the additive white Gaussian noise (AWGN), Rician fading, Rayleigh fading, and power amplifier AWGN channels.
- We capture the simulated block error rate (BLER) for transmitter and receiver models measured over the generative channel model and demonstrate that this measurement correlates well with the BLER measured over the true channel environment, thereby showing that the simulated BLER is suitable for use as the training stopping criteria and for monitoring of the learning process.
- Finally, we demonstrate that the performance of the resulting transmitter and receiver models is equivalent to or better than the end-to-end model that is trained with an assumed channel model. This is shown for AWGN, Rician fading, Rayleigh fading, and non-linear power amplifier distortions over AWGN simulated channels, thereby matching the performance of more complex OAL methods that compare against the end-to-end model in the literature.

In this article, we present the background for end-to-end learning and related work in Section 2. In Section 3, we describe the system model and our proposed approach. We present and discuss results for the proposed approach compared with the end-to-end method in Section 4. In Section 5, we discuss limitations and simplifying assumptions for the proposed method and describe how these may be addressed in Section 6.2, which describes avenues for Future Work. Finally, we summarise our findings and conclude our paper in Section 6.1.

2. Background and Related Work

The most commonly cited motivation for the use of DL in training a wireless communication system is for its potential as a data-driven method to jointly optimise both the receiver and transmitter with respect to the distortions produced from the channel [1,4,5,7,9–14]. This motivation has spurred much investigation into the practical considerations required to realise the goal of automated design. Notably, the end-to-end design was first presented in [1], which demonstrated the application of the autoencoder (AE) model to the end-to-end joint optimisation using backpropagation for the transmitter and receiver over an assumed channel. The AE structure is divided into an encoder or transmitter component, a differentiable channel transfer function, and a decoder or receiver component. It is demonstrated to learn an encoding that can produce a BLER similar to the conventional Hamming(7,4) code over the AWGN channel [1]. The backpropagation training of wireless communications systems suffers a significant flaw, however, and that is the requirement for end-to-end differentiation must also assume a differentiable channel transfer function. This limitation prevents the design method from being applied in physical channel environments.

The simplest way to address this limitation is to take a two-step approach: first, training the end-to-end system offline, and second, performing tuning of the receiver model in the true channel environment. This procedure is demonstrated in [9], with a more realistic channel function that includes upsampling, timing, phase, and frequency offsets. Incorporating these additional distortions in the channel function required additional design considerations in the receiver model, which included a data preprocessing step to slice windows of the incoming signal, a phase estimation, and general feature extraction layers whose outputs were concatenated to feed into the receiver classifier [9]. The transmitter and receiver architectures were trained end-to-end, and the receiver was tuned post-deployment in both simulated AWGN and physical channels. The performance of the AE did not quite match the conventional differential quadrature phase-shift keying (QPSK) modulation but did demonstrate the first practical application of end-to-end training to OAL. Joint optimisation of both the transmitter and receiver models remained elusive, however, since only the receiver benefited from tuning in the deployed channel environment.

Gradient approximation methods were developed to enable optimisation for both the transmitter and receiver in OAL without prior knowledge of the channel. Two notable approaches were developed, the first being derived from simultaneous perturbation stochastic approximation (SPSA) [2] and the second based on Reinforcement learning (RL) policy gradient methods [15]. Both methods require that the transmitter outputs are perturbed multiple times to sample the loss from the receiver at several small displacements around the transmitter outputs [2,15]. The SPSA method requires more sampling than the latter method and does not scale well to longer messages or more complex transmitter models [3]. Both approaches did, however, demonstrate the feasibility of the method and achieved performance equivalent to the joint end-to-end approach in AWGN and Rayleigh fading channels. Subsequent work has advanced the use of the RL-based approach with application to concatenated coding and demonstrating good performance on longer message sequences, which addresses the short message limitation for symbol-wise classification in end-to-end learning [10]. However, reliance on the feedback channel increases channel use and the vulnerability to data poisoning during training, and multiple forward passes through the transmitter in each single training epoch can be avoided with an appropriate proxy channel model.

A DL channel model can be applied to learn the physical channel environment directly from observations without assuming a model for the true channel. Once trained, the channel model acts as a proxy to support backpropagation in the end-to-end training for transmitter and receiver models. GAN training methods have been adopted for their ability to approximate a distribution given noisy inputs. A variational AE generator was applied in [5] to receive transmitter outputs and approximate the channel distribution for several channels. The variational AE generator maps the transmitter symbols to the parameters for an internal normal distribution and uses samples from the inner distribution to map into the channel distribution. This method enables the model to approximate the stochastic quality of the channel [5]. The method is shown to approximate several channels, including AWGN, a non-Gaussian Chi-squared channel effect, and a non-linear channel over AWGN, which includes a hardware amplifier [5]. While this article demonstrated the potential application for modelling channels using the GAN, it did not consider how to apply the resulting channel model in the end-to-end training regime.

Instead of sampling with a variational AE, the context information produced by transmitting pilot symbols was applied to help the generator approximate the channel function in [4]. The resulting conditional GAN is trained on simulated AWGN and Rayleigh fading channels, and then used as a proxy for the true channel to train the transmitter and receiver [4]. The resulting performance was very close to the Hamming(7,4) code on the AWGN channel and was similar to coherent detection in the Rayleigh fading channel [4]. Refs. [4,5] train the AE with the adversarial learning algorithm where a separate discriminator aims to differentiate between true and generated samples and the generator aims to fool the discriminator into misclassifying generated samples [4]. However, one problem in adversarial training is that the generator model can suffer from mode collapse, where it confines generated results to a smaller area of the broader distribution to consistently fool the discriminator and subsequently fails to perform generalisation in modelling the extent of the target distribution [16].

The Wasserstein generative adversarial network (WGAN), which modifies the adversarial loss function, is proposed to improve training stability and address the issue of mode collapse for GAN training [17]. A WGAN model is trained on the receiver side without the need for continuous feedback in [7]. The target data set is first constructed by using a pre-trained transmitter to send a batch of transmissions through the channel. Once the batch has been collected, the WGAN can be trained with adversarial learning, and the resulting generator can be used to train a transmitter and receiver end-to-end. Instead of applying symbol-wise decoding, the approach used bit-wise decoding in a manner similar to [10]. The authors demonstrated one of the first instances where the GAN approach was applied in a physical channel to train the transmitter and receiver. However, when experimenting with the more dynamic time delay channel, the WGAN did not converge due to mode collapse, indicating that generative methods are challenged when learning more complex channels [7].

A conditional GAN that is trained on both transmitter symbols and received pilot symbols is proposed in order to generate more complex time-varying channel distributions in [11]. The method extends the work in [4] to longer codes using convolutional neural network (CNN) layers and proposes an iterative training algorithm for transmitter, GAN, and receiver. By including the pilot symbols as well as the transmitter symbols, the generator model is able to more closely match the channel effects observed during training [11]. Evaluation of the resulting system in simulated AWGN, Rayleigh fading, and frequency-selective fading channels demonstrates similar performance to that of an end-to-end AE trained with an assumed channel. However, the transmitter, channel, and receiver models are trained in an iterative manner [11], indicating a high channel usage during the training procedure similar to the RL method.

Rather than generating the channel distribution directly, the authors in [13] use a residual connection to learn the distribution of the differences between transmitter symbols and the received symbols output by the channel. The method residual aided generative adversarial network (RA-GAN) is trained on simulated channel data via an iterative training scheme and evaluated against a GAN-based model [13]. Evaluation in the AWGN, Rayleigh fading, and a ray-tracing-based channels demonstrates performance close to the optimal end-to-end AE training scheme and is close to the performance for both WGAN- and RL-based methods [13]. The approach simplifies the structure of the GAN, as well as introduces an additional regularisation term. However, the approach shares the same disadvantage as the other GAN-based training methods.

Each of the GAN-based methods requires a separate discriminator neural network that is used to train the generator during the adversarial training procedure. Adversarial training is a two-step procedure where the discriminator is first trained to classify true channel observations versus the generated samples, and secondly, the discriminator is used to train the generator to produce samples closer to the true observations [11]. After training the channel model, the discriminator is discarded. However, if considering training OAL on embedded IoT devices, there will be limitations to the capabilities of hardware platforms, unlike host driven software defined radio (SDR). It is preferable to reduce the number of models, which each requires training iterations; therefore, a single-channel model that can accurately approximate the channel distribution is preferable.

Difficulty with training stability for the GAN model has been quoted as a motivation for the different variations that have been applied in the literature [4,7,13]. An alternate single-channel model, the diffusion-denoising probabilistic model (DDPM). is adopted in [14], primarily to address the issue of mode collapse in the GAN method and because it has shown excellent performance in the image generation domain. The DDPM learns the parameters for the variance of a forward noising process where Gaussian noise is repeatedly added starting from the original input, and a reverse process which learns to restore the original data from the noise [14]. However, the denoising procedure is slow, requiring multiple recursive steps, hence a variation of the approach denoising diffusion implicit model (DDIM) is proposed to trade-off between accuracy and time [14]. Two approaches of training are adopted for comparison: the pre-trained approach trains the channel generator model before using it in the end-to-end training procedure, and the iterative approach interleaves training of channel generator, transmitter, and receiver [14]. Evaluation of the trained transmitter and receiver models is carried out with a $K = 4$, $N = 7$ code in the simulated AWGN, Rayleigh fading, and non-linear amplifier AWGN channels [14]. Pre-training was demonstrated to have the closest performance to the original end-to-end training method, and 50 iterations for the DDIM method was shown to be a good trade-off between accuracy and speed in comparison to the DDPM approach [14]. While diffusion models have demonstrated excellent generative capabilities in the image domain, the number of iterations to perform denoising adds to the latency during training, which is a disadvantage for the application of this approach to OAL. The advantage of the GAN is that after training, the channel can be simulated with a single forward pass. However, the training complexity due to adversarial learning against a discriminator model is the primary limitation for GAN-based methods in OAL. Therefore, a generative model that does not require multiple passes to reconstruct the signal and that supports a simple training regime is desirable for applications that may operate on embedded devices over a physical channel environment.

MDNs combine conventional neural networks with a mixture density model to learn an underlying generative mapping between input and target data [18]. The MDN trains a neural network to approximate general distributions by learning the parameters for a Gaussian mixture model [18]. In this manner, it is trained using conventional supervised learning without the need for a discriminator or multiple applications of noise and is a much simpler modeling framework than the GAN or diffusion-denoising models. A standard network can be seen as learning the mean of the target mapping through the least-squares loss, and the MDN instead models the parameters for the distribution of multi-valued continuous target variables [18]. This advantage over standard neural networks makes the MDN suitable for use in optimization problems, which may include non-unique solutions for different parameters [19]. This has led to the application of MDN to parameter estimation for inverse problems [20–22] and to simulation of physical processes [23].

Parameter estimation in the wireless environment is especially challenging due to noise and fading as well as other distortions such as timing, frequency, and phase offsets. However, the MDN has been demonstrated to enable accurate estimation for localisation of wireless sensor network devices in an environment featuring both AWGN and fading effects in [20]. The MDN has also been demonstrated to provide accurate approximation for the distributions of latency measurements taken in a 5G wireless AWGN environment [21]. In a related domain, the MDN was applied to the estimation of direction of arrival for acoustic signals also within an AWGN environment, and was shown to capture an accurate model of the uncertainty due to the channel [22]. In the radar domain, the MDN is demonstrated as an effective data-driven method to approximate radar sensor measurements for distance, velocity, and orientation of a moving vehicle [23]. In this scenario, a transmitted chirp signal is distorted by channel perturbations and noise as well as fading and the Doppler effect [23].

In this article, we propose a method for OAL without feedback, thereby reducing the channel use and opportunity for data poisoning attacks. A MDN channel model is trained by observing transmitted random uniform noise over the true channel environment. The MDN can be trained in a supervised manner to approximate the true channel distribution without use of a discriminator for adversarial training and is able to learn without the need of multiple forward passes or repeated noise correction in each epoch.

3. Methodology

3.1. System Model

In our work, we assume a single input single output (SISO) wireless communications system. A K bit binary message M is coded with an N bit code and modulated at the transmitter to produce a set of complex transmitter symbols $z \in \mathcal{C}$. Experiments are carried out with $K = 8$ bits and $N = 8$ symbols. The transmitter symbols are transferred over the wireless channel, which we simulate as a transfer function $r(t) = h(z(t))$. The channel adds noise and other perturbations such as fading. In this article, training is carried out at a fixed signal to noise ratio (SNR) of 6 dB, and evaluation is performed over the SNR range of 0 dB to 15 dB. The receiver is responsible for detecting the signal, correcting distortions, demodulation, and decoding to produce an estimate of the original message $\hat{M}$. In our system, we assume perfect synchronisation; therefore, we do not add additional effects such as time delay, phase, or carrier frequency offset. The set of channels that are applied in this article are described in Section 3.5.

The developed AE-based transmitters and receivers learn to produce an uncoded modulation. Therefore, we include the BLER for uncoded binary phase shift keying (BPSK) to provide a reference for the optimal performance of an uncoded modulation. The difference in the performance is due to the ability of the AE to learn to utilise the entire in-phase and quadrature (IQ) space in the learnt constellation as opposed to using only two symbols available to BPSK modulation.

3.2. Joint End-to-End Approach

The joint end-to-end approach, based on the AE from [1], is depicted in Figure 2. This approach is trained end-to-end and incorporates a differentiable channel function $h(y)$ as part of the model. The transmitter inputs consist of the one-hot encoded vector for the K bit message M. The one-hot encoding indicates the ith message as a one in $i \in 2^K$ index positions, where all other positions $j \neq i$ are set to 0. The output at the receiver is a vector of 2^K probabilities $p(y|r)$ conditioned on channel symbols r where $\sum_{i=1}^{2^K} p(y_i|r) = 1$. This is facilitated by the softmax activation $p(y_i|r) = exp(l_i) / \sum_j^{2^K} exp(l_j)$ where l is learnt by the receiver neural network. The index for the maximum probability is mapped to the corresponding index of the original message $M_{\text{index}} = \text{arg max } p(y|r)$. Under this regime, the end-to-end model is trained against the cross-entropy (CE) loss shown in Equation (1). $p(y_{true})$ is represented as the one-hot encoding for the true messages and $p(y|r)$ is softmax output produced by the receiver. In our work, we consider this joint model the baseline AE model, which has assumed knowledge of the channel environment, and we compare our proposed method to this model.

$$\mathcal{L}(p(y_{true}), p(y|r)) = -\sum_{i=1}^{2^K} p(y_i) \log p(y_i|r) \tag{1}$$

In the literature for OAL methods, the joint end-to-end AE based on [1] serves as the baseline comparative method. This is because, under simulation, the assumed channel function provides the joint end-to-end AE with complete information of the simulated environment and hence provides the optimal performance for the DL-based method. To demonstrate the effectiveness of the proposed method our aim is to demonstrate equivalent performance, since our proposed method does not have complete information about the channel, it must rely on training a proxy model of the true channel environment to learn an optimal constellation for that environment.

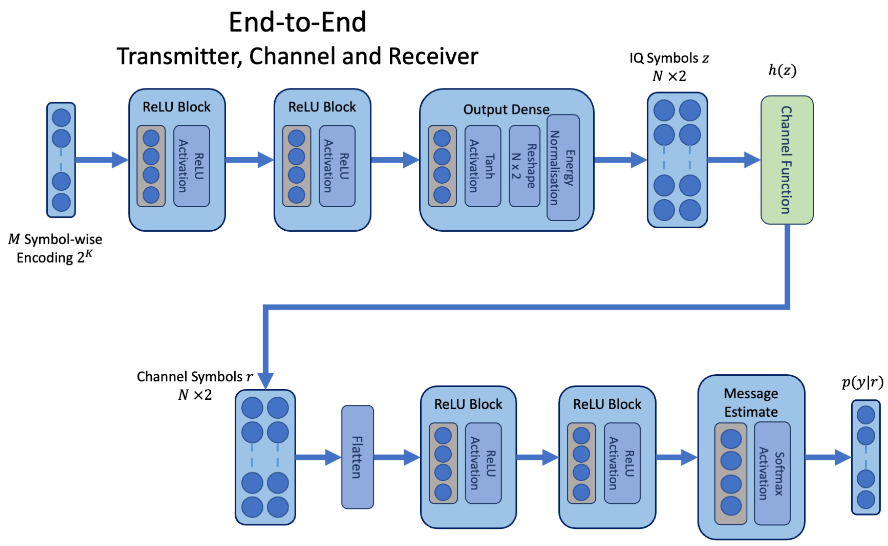


Figure 2. The end-to-end network architecture where an assumed channel transfer function is defined as a layer within the network architecture.

3.3. Proposed Approach

The transmitter and receiver blocks in our proposed approach differ from the original AE in [1]. Instead of dense blocks, we define a residual block with skip connections between the dense units, illustrated in Figure 3. The skip block in our architecture (shown in Figure 3) consists of three dense blocks consisting of linear units, batch normalisation [24] and a swish activation [25]. The first block scales incoming features so that they have a compatible dimension for addition to the output of the final block. Skip connections, also known as residual connections, mitigate vanishing gradients in deeper networks and are indicated to form an ensemble of models by combining multiple paths through the network [26]. Any number of skip blocks may be arranged in sequence in the network architecture. In our model, we typically used one skip block per transmitter and receiver network. Our choice of the swish activation is related to our choice of skip connections. The ReLU activation is known to suffer from a vanishing gradient due to its exclusion of negative values [27]. We chose the swish activation function to help mitigate the vanishing gradient and complement the use of the skip blocks to aid in promoting learning during backpropagation. Experimentally we have found the swish activation to outperform ReLU activations, as indicated in [25]. Since the swish activation is unbounded for positive values, batch normalisation is applied to reduce the impact of extremes in the activation values.

The L symbols, output by the transmitter, are scaled to emulate the energy constraint of the transmitter hardware such that $||x||_2^2 \leq 1$, shown in Equation (2). The tanh activation function is applied at the output of the transmitter to ensure that the learnt transmitter symbols remain within the range $[-1, 1]$. Instead of an assumed channel function, back-propagation is enabled by the channel model, which is trained to approximate the true channel during the proposed training procedure (described in Section 3.4). The channel model connects transmitter and receiver models and acts as the proxy for the true channel to allow training to take place on the receiver side. This allows the training to occur without the need for feedback over the true channel, thereby reducing the opportunity for data poisoning during the training of the transmitter and receiver. Once trained, the transmitter weights can be transferred to the origin transmitter side to send messages across the true channel. Tables 1 and 2 indicate the respective network dimensions for the transmitter and receiver.

$$z(t) = \frac{x(t)}{\sqrt{\sum_{i=1}^{L} x(i)^2 / L}} \quad (2)$$

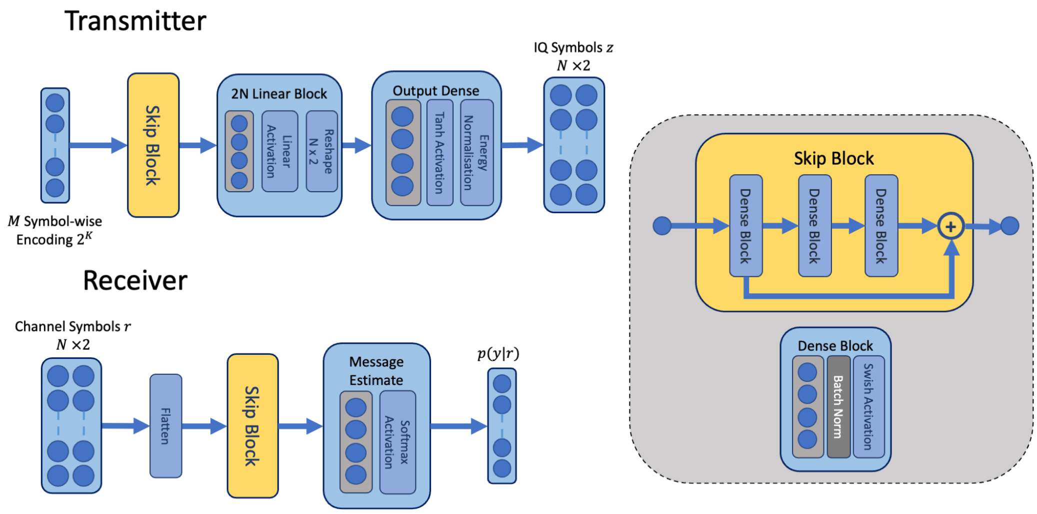


Figure 3. The architecture of the transmitter and receiver networks containing the dense residual skip block for feature extraction.

Table 1. The transmitter consists of four groups, input, skip block, a linear transformation, and an output block. The number of units is specified for the dense layers, batch normalisation, and swish activation preserve the same dimension of output as produced by the dense layer. The model was trained to map an uncoded message of $K = 8$ bits to $N = 8$ IQ symbols.

Layer	Units Uncoded 8 Bit	Group
Input layer	2^K	Input
Dense layer	512	Skip block
Batch normalisation	-	
Swish activation	-	
Dense layer	64	
Batch normalisation	-	
Swish activation	-	
Dense layer	512	
Batch normalisation	-	
Swish activation	-	
Dense layer	$2N$	$2N$ linear block
Linear activation	-	
Reshape $[N, 2]$ layer	-	
Dense layer	2	Output $[N, 2]$
Tanh activation	-	
Energy normalisation	-	

To train the transmitter and receiver, we apply a channel MDN model. The channel MDN model is trained in a supervised manner against observations of noise transmitted through the true channel. Unlike the GAN, it does not require a separate discriminator model and does not require multiple denoising steps in comparison to the diffusion modelling approach. The resulting channel model is combined with the transmitter and receiver during an end-to-end learning phase, where it emulates the true channel environment. The MDN model estimates parameters for the mean and standard deviation $\theta = \{\mu_j, \sigma_j\}$

and mixing coefficients ϕ_j for $j = 1 \dots J$ Gaussian distributions for each individual symbol $z(t)$ in L time-steps [18]. The resulting mixture of Gaussian distributions is combined to generate a probability density over the channel outputs $p(r(t)|z(t))$ (Equation (3)). In our implementation, each symbol may have a different mean and standard deviation, which are produced by the main path of the network consisting of a skip block and dense linear block illustrated in Figure 4.

Table 2. The receiver network consisted of three groups for input, feature learning (skip block), and output. The dimensions of units are shown for each dense layer with subsequent layers producing the same shape output as the preceding dense layer. The receiver was trained to map $N = 8$ IQ symbols to the original $K = 8$ bit message.

Layer	Units Uncoded 8 Bit	Group
Input layer	$[N, 2]$	Input
Flatten layer	-	
Dense layer	512	Skip block
Batch normalisation	-	
Swish activation	-	
Dense layer	64	
Batch normalisation	-	
Swish activation	-	
Dense layer	512	
Batch normalisation	-	
Swish activation	-	
Dense layer	2^K	Output
Softmax activation	-	

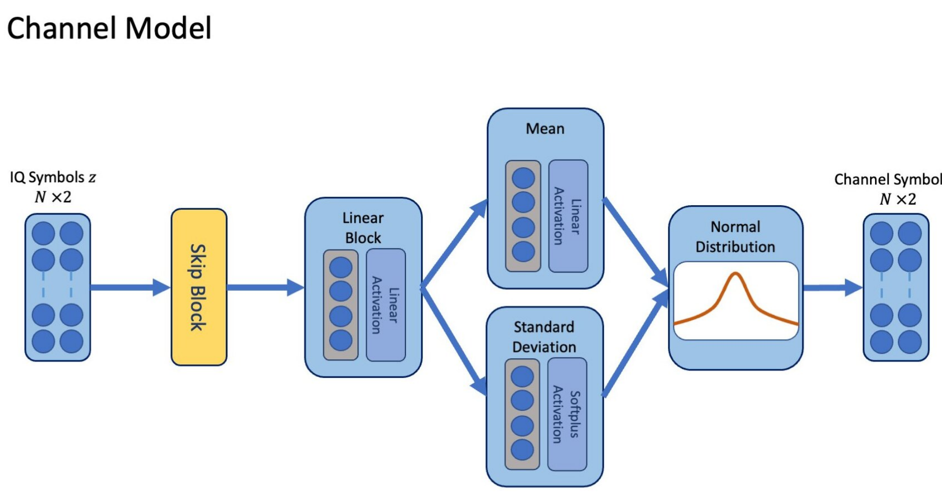


Figure 4. The channel model estimates parameters for the mean and standard deviation of a normal distribution for each transmitted symbol, and the estimate for channel effects is sampled from the resulting distribution.

Our model is a simplification of the MDN since we use only one Gaussian distribution and do not model coefficients [18]. However, we do learn separate mappings for each time-step from $z(t)$ to θ. While it is possible to extend the modelling approach to include more than one set of Gaussian distributions, we found that estimating an individual mean and variance for each IQ symbol was sufficient for the set of channels used in the evaluation.

$$p(r(t)|z(t)) = \sum_{j=1}^{J} \phi_j(z(t)) \mathcal{N}\left(\mu_j(z(t)), \sigma_j^2(z(t))\right) \quad (3)$$

The network is trained by minimising the negative log-likelihood (NLL) loss, shown in Equation (4), where $r(t)_{\text{true}}$ are the true channel responses, $z(t)$ are the transmitter symbols, and θ are the distribution parameters learnt by the MDN. A linear activation is applied to the estimate for the mean, and a softplus activation [28] is added with a small positive constant for the standard deviation. While this model is simpler than other generative approaches such as the GAN and diffusion models, it performs well in enabling the transmitter and receiver to learn modulation and coding that produces equivalent or better BLER when compared with the end-to-end learning approach. Table 3 lists the dimensions for each of the layers in the channel model.

$$\mathcal{L}(r(t)_{true}, z(t), \theta) = -\ln\{p(r(t)_{true}|z(t), \theta)\} \tag{4}$$

Table 3. The channel model receives as input the transmitter symbols $z(t)$ and learns to approximate the distribution for the true instantaneous channel function $r(t)$. The final layers learn the parameters for the mean and standard deviation of a normal distribution around each IQ symbol in $r(t)$.

Layer	Units Uncoded 8 Bit	Group
Input layer	$[N, 2]$	Input
Dense layer	512	Skip block
Batch normalisation	-	
Swish activation	-	
Dense layer	64	
Batch normalisation	-	
Swish activation	-	
Dense layer	512	
Batch normalisation	-	
Swish activation	-	
Dense layer	512	Skip block
Batch normalisation	-	
Swish activation	-	
Dense layer	64	
Batch normalisation	-	
Swish activation	-	
Dense layer	512	
Batch normalisation	-	
Swish activation	-	
Dense layer	$[N, 4]$	Distribution Parameters
Mean branch	$[N, 2]$	
Standard Deviation branch	$[N, 2]$	
Mean Linear activation	-	
Standard Deviation Softplus activation	-	
Sample Normal distribution	$[N, 2]$	Output

3.4. Training Procedure

An overview of the training procedure is illustrated in Figure 5, where in stage 1, the initialisation of the training procedure requires that both an origin transmitter and remote receiver share the same random seed. This is used to draw continuous IQ samples of desired block length K from the uniform distribution $S \sim U([-1, 1])$. We emphasise that the random sequence S is not an information-carrying modulation. In stage 2, the random sequence is transmitted from the origin transmitter to the remote receiver, producing channel symbols R. A batch size of 128 blocks is collected prior to training the channel model. Backpropagation is applied in stage 3 to train the remote channel model against the true channel symbols R using the NLL loss. Stage 4 performs end-to-end training of the transmitter and receiver on the receiver side using the trained channel model, without the need for a feedback channel.

In stage 4, the weights of the channel model are frozen so that they are not updated. A batch size of 32 random K bit message blocks M is generated prior to performing backpropagation on the end-to-end version of the model, with the channel model acting as the true channel proxy. The procedure is repeated until convergence, which is indicated by the validation loss (a validation batch size of 32 random K bit message blocks is used to measure this loss for the stopping condition). This process is repeated for 1600 steps in each training epoch with up to a maximum of 300 epochs. After a single epoch, the simulated channel block error rate ($\mathrm{BLER}_{\mathrm{sim}}$) is calculated against the proxy channel model for monitoring purposes. In our experiments, we have also calculated BLER against the true channel to measure the correlation between the $\mathrm{BLER}_{\mathrm{sim}}$ and the BLER. We observe that while the $\mathrm{BLER}_{\mathrm{sim}}$ has higher variance, it is well correlated with the BLER and is a suitable indication of expected model performance at the current SNR of the channel. In our experiments, we trained on a simulated channel at an SNR of 6 dB.

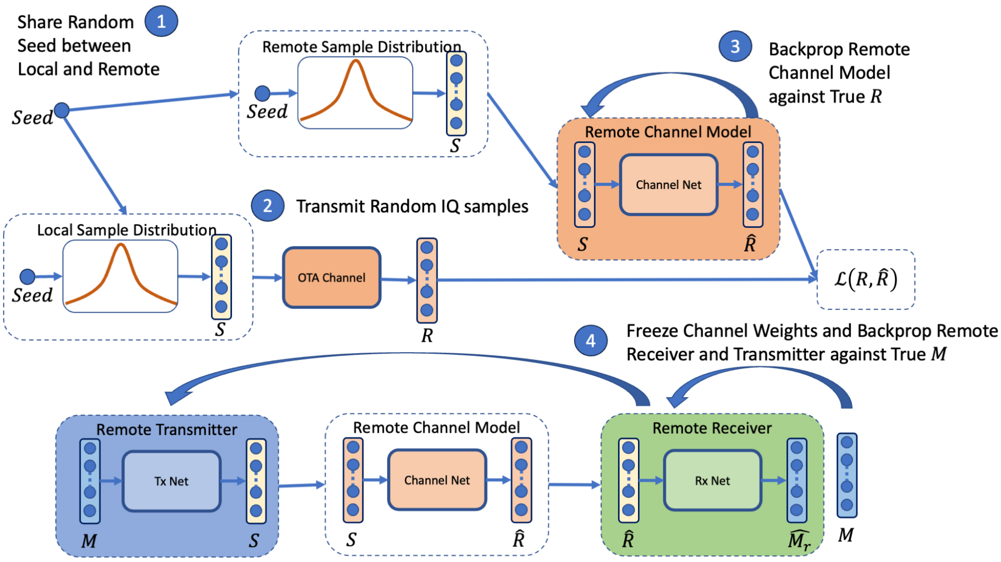


Figure 5. Schematic view of the proposed OAL procedure, without feedback. Stage 1 samples random values from the uniform distribution using a shared seed. Step 2 transmits the random values S over the true channel to receive channel symbols R. Step 3 trains the remote channel model on input S and back propagates against true channel symbols R. In Step 4, the channel model weights are frozen, and training via backpropagation for message M is performed using the remote channel model as a proxy for the true channel.

3.5. Simulated Channel Environments

To investigate the performance of the model, we train on four instantaneous channel functions, an AWGN channel, a Rayleigh fading channel, a Rician channel with Rician factor equal to 4, and a non-linear power amplifier with an additive white Gaussian Noise (PA-AWGN) channel. Each of these functions includes an additive noise component as shown in the AWGN channel Equation (5).

$$r(t) = z(t) + n(t) \tag{5}$$

The Rayleigh and Rician fading channels scale the transmitter symbols $z(t)$ with fading coefficients $a(t)$ Equation (6). However, they each differ in how the fading coefficients are calculated.

$$r(t) = a(t)z(t) + n(t) \tag{6}$$

In the Rayleigh fading channel, the fading coefficients are drawn from a complex standard normal distribution $a \sim \mathcal{C}N(0,1)$ and their argument is scaled and multiplied with a phased waveform $a(t) = \frac{1}{\sqrt{2}}|a|e^{j\psi}$, in our case we assume a zero phase $\psi = 0$.

The Rician fading channel function $r(t)$ in Equation (6) draws its scaling coefficients from a complex normal distribution with parameters μ and σ, $a \sim \mathcal{C}N(\mu, \sigma^2)$. The mean $\mu = \sqrt{\frac{K}{(2(K+1))}}$ and standard deviation $\sigma = \sqrt{\frac{1}{(2(K+1))}}$ are parameterised by the Rician factor K, which we define as $K = 10$. The lower the value for K, the Rician fading appears to become similar to Rayleigh fading, and for higher values of K, Rician fading resembles AWGN.

In the PA-AWGN channel, we apply a solid-state high-power amplifier (SSPA) to translate transmitter symbols prior to the AWGN channel. Equation (7) shows the Rapp model [29] with parameters for the limiting output amplitude A_0, amplifier gain ν, and smoothness p. In our experiments, these are set to $A_0 = 1$, $\nu = 1$, and $p = 5$. The transmitter symbols are then transformed by the amplifier function, as shown in Equation (8).

$$g(A) = \nu \frac{A}{\left(1 + \left[\left(\frac{\nu A}{A_0}\right)^2\right]^p\right)^{1/2p}} \tag{7}$$

$$z'(t) = g(|z(t)|)e^{j\angle z(t)} \tag{8}$$

Since in this article we simulate the channel, we parameterise the function with the ratio of energy per information bit to the noise power spectral density E_b/N_0 in dB and a parameter for the code rate K/N. We use the code rate to convert to the ratio for the energy per symbol to noise power spectral density E_s/N_0 dB $= E_b/N_0$ dB $+ 10log_{10}(K/N)$ and use the linear form $E_s/N_0 = 10^{E_s/N_0 \text{ dB}/10}$ to estimate the separate components $E_s = \sum_{t=1}^{L} z(t)^2/L$ and $N_0 = E_s/(E_s/N_0)$. The noise variance $\sigma^2 = N_0/2$ is then used to draw the complex Gaussian noise parameter $n(t) \sim \mathcal{C}N(0,\sigma^2)$. Once the noise is determined, it can be applied to the channel transfer function.

We trained and evaluated our proposed method and the joint end-to-end method against each of these channels. The joint approach is based on the model defined in [1] and includes the instantaneous channel transfer function as part of the network architecture. In this approach, there is no requirement for an iterative training algorithm, and the training is performed by backpropagation over a maximum of 300 $\times$ 12 epochs, with a batch size of 32 $K = 8$ bit messages. The Adam optimisation algorithm [30] is applied in both approaches, and we leverage stochastic weight averaging (SWA) [31] every 10 epochs with a cyclical learning rate schedule [32] having a minimum learning rate of 0.0001 and maximum of 0.001.

4. Results and Discussion

While the baseline end-to-end joint and proposed iterative methods take different approaches to training, once trained, the transmitter and receiver can be separated from the end-to-end model and deployed separately for testing. In the iterative method, the channel model is not required for deployment and is used only during training. Both approaches are evaluated by transmitting generated random $K = 8$ bit message blocks and transmitting over each of the simulated channel transfer functions. The BLER is calculated for each block at varying SNR between 0 to a maximum of 15 dB. In this section, we present results for both methods, as well as the uncoded BLER maximum likelihood decoding performance.

The performance of both methods under each channel is presented in Figure 6, and is compared with uncoded BPSK for reference. Even though the proposed iterative method has been trained on a generated model, while the joint method is trained with an assumed channel function, there is very little difference between the performance of both. The PA-AWGN channel is an exception, however. The proposed method outperforms the joint method, which appears to have an error floor in higher SNR. Each of the DL methods achieves gains over the uncoded BPSK modulation. This is because the uncoded BPSK modulation represents the 8 bit sequence with 8 symbols, each chosen from one of two constellation points. For example, $-1 + 0j$ for 0 and $1 + 0j$ for 1. Whereas the DL methods can map each one of the 2^K messages to any arrangement of 8 symbols in the IQ space. The DL methods learn this mapping by minimising the error in message recovery subject to the distortions introduced by the channel.

The joint end-to-end model has been trained with full information of the simulated channel environment, due to the assumed channel layer. In contrast, our proposed approach trains a separate channel model to act as a proxy for the true channel environment, given observations of random noise. The RL, GAN and diffusion approaches outlined in the literature compare solutions with variants of the canonical joint end-to-end learning method [4,7,9,11,13–15]. This is to demonstrate equivalent or better performance against the model, which is trained with the assumed channel function. Doing so indicates that the method learns an optimised code based on the observations without prior knowledge of the channel. The BLER performance for our proposed approach indicates that the resulting channel model provides an accurate representation of the true channel environment. This approximation enables the transmitter model to learn an optimised code for the target channel environment.

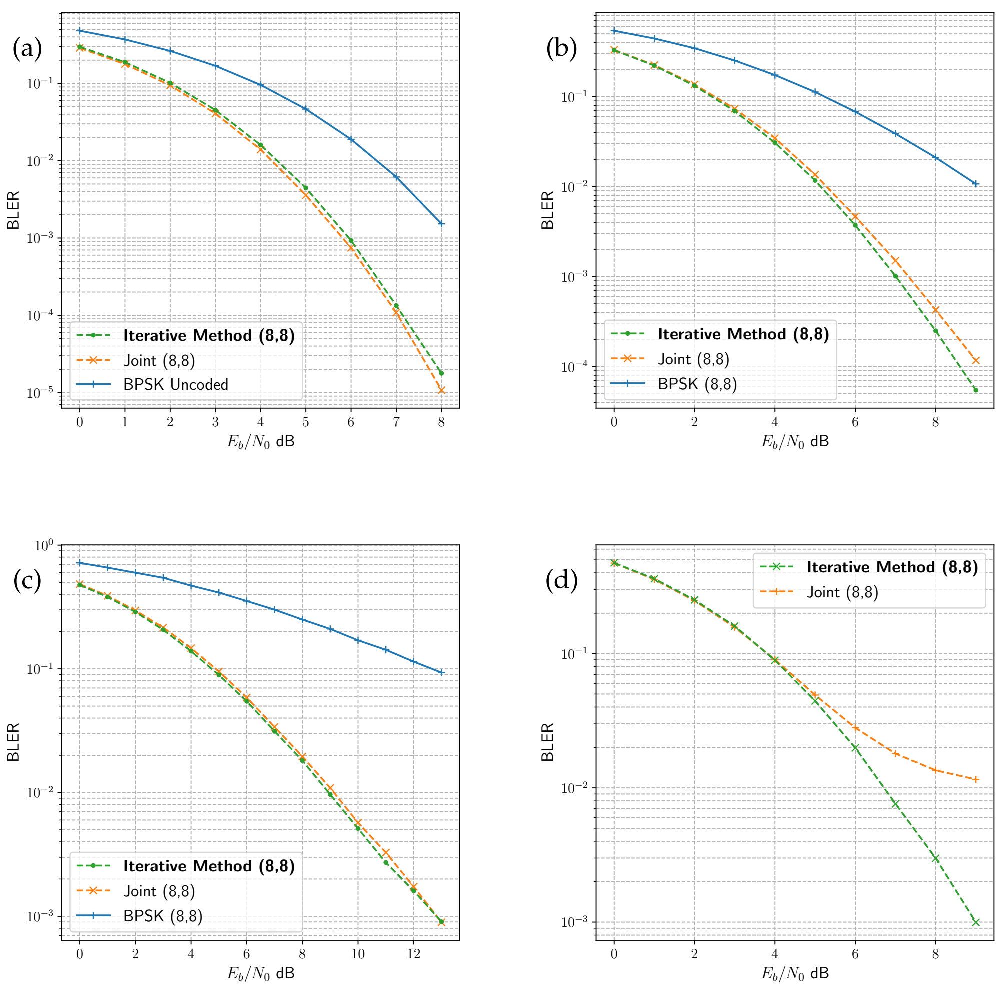


Figure 6. Comparison of BLER performance in the four channel environments. Uncoded $K = 8$ bit BPSK modulation is compared with the joint and the proposed iterative method in the (**a**) AWGN channel, (**b**) Rician fading channel, (**c**) Rayleigh fading channel, and joint and iterative methods are compared in (**d**) the PA-AWGN channel.

During the training of the channel model, the origin transmitter samples are drawn from the random uniform distribution $s(t) \sim U([-1,1])$ prior to transmitting over the instantaneous channel function. The channel model does not learn from an information-carrying modulation, as such it does not learn unique features specific to a given waveform. While this could be a disadvantage, the BLER performance indicates that the channel model provides a suitable approximation that enables the transmitter and receiver to jointly learn an appropriate representation for the transmit symbols. In our evaluation, we review the channel effect on a BPSK modulation and compare this with the estimates produced by the channel model. The channel model is able to approximate distributions of the instantaneous channel as shown in Figure 7. The intention of training on uniform IQ samples is to prevent transmission of an intelligible information-carrying signal during the training procedure. The resulting bimodal distribution for each channel function with the BPSK modulation is approximated well by the trained channel model, which produces a mixture of Gaussians with different scales and locations corresponding to the two modulation symbols.

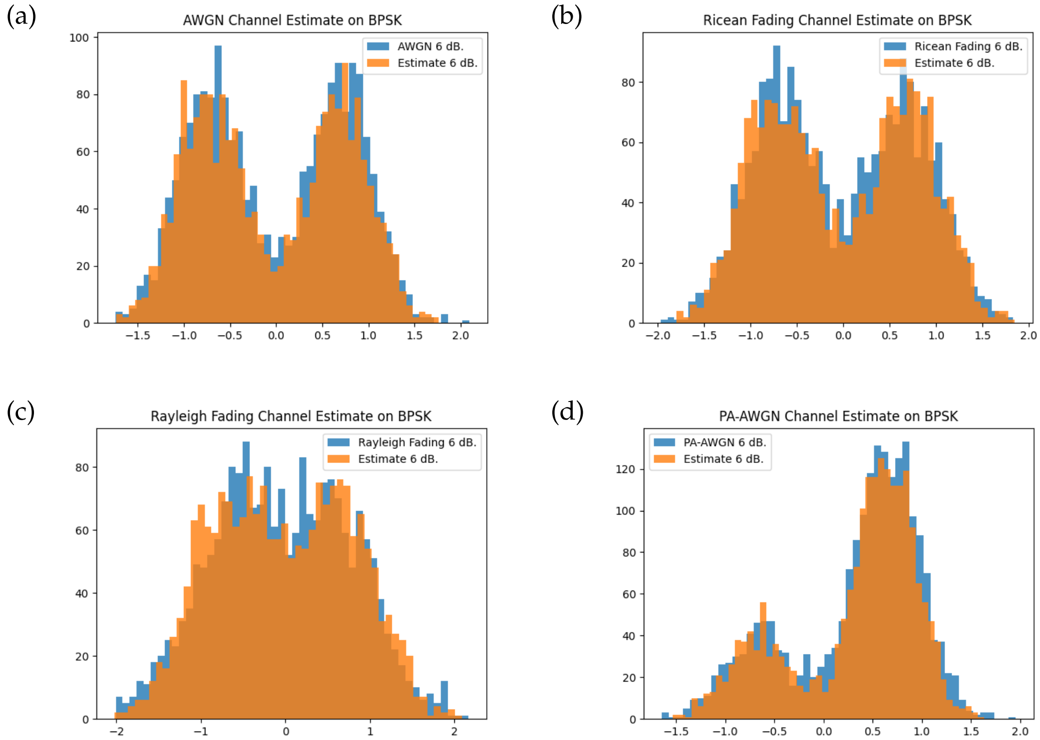


Figure 7. Histogram of channel symbols for each instantaneous channel function and the trained channel model at training SNR of 6 dB for random BPSK modulation. Comparisons are shown for the (**a**) AWGN channel, (**b**) Rician fading channel, (**c**) Rayleigh fading channel, and (**d**) PA-AWGN channel.

The transmitter model, however, does not learn a conventional modulation; instead, the transmit symbols make use of the IQ space more broadly. Figure 8 shows the histogram for the instantaneous channel transfer function and the approximation given by the channel model when provided with the learnt transmitter symbols. The channel model approximation is close to that of the true distribution when presented with a non-uniform modulation.

The question of when to stop training often relies on monitoring a performance metric such as the validation loss, and once the metric ceases to decrease after a fixed number of steps, the training cycle ceases. However, when the intention is to carry out training without feedback over the true channel, training metrics may no longer be reliable for determining whether the end-to-end system is learning under the true channel conditions. Outside of the negative log-likelihood loss for the channel, it is desirable to be able to monitor a performance metric which is a good indicator of the training progress of the end-to-end system. Our intuition is that if the channel model is learning an accurate representation of the true channel transfer function, the BLER produced by evaluation of transmitter and receiver via the channel model should reflect the BLER that would be produced over the true transfer function. Evaluation of the transmitter and receiver was performed on both the instantaneous channel transfer function as well as the channel model at the end of each epoch. Figure 9 shows the monitored value of the BLER during training at the fixed SNR of 6 dB. We note that, in general, the BLER corresponds well with that recorded on the true channel, apart from the Rician fading channel, where the simulated BLER is lower. However, the error signal correlates well in each channel and serves as a suitable proxy measure during training (Table 4). It is also worth observing that the variance of the BLER differs between the true and simulated channels. This is more visible in the Rician and Rayleigh fading channels, which have a larger number of training epochs.

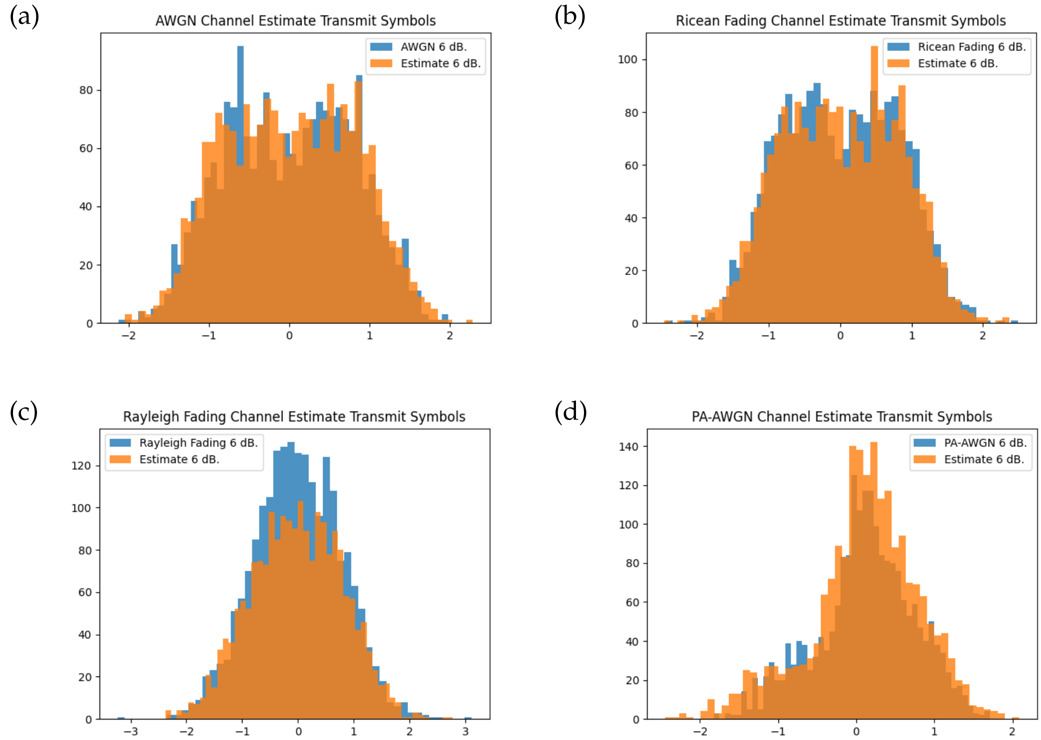


Figure 8. Histogram of channel symbols for each instantaneous channel function and the trained channel model at training SNR of 6 dB for $K = 8$ bit messages transferred through the transmitter model. Comparisons are shown for the (**a**) AWGN channel, (**b**) Rician fading channel, (**c**) Rayleigh fading channel, and (**d**) PA-AWGN channel.

Table 4. Pearson ρ_c and Spearman's rank ρ_s correlation coefficients for the BLER produced on true and simulated channels. The high correlation as well as the error curves indicates a suitability for the simulated channel model to act as a proxy for performance monitoring during training as well as acting as a metric for the training stopping condition.

Channel Type	ρ_c	ρ_s
AWGN	0.94	0.89
Rician Fading	0.94	0.73
Rayleigh Fading	0.88	0.66
PA-AWGN	0.99	0.93

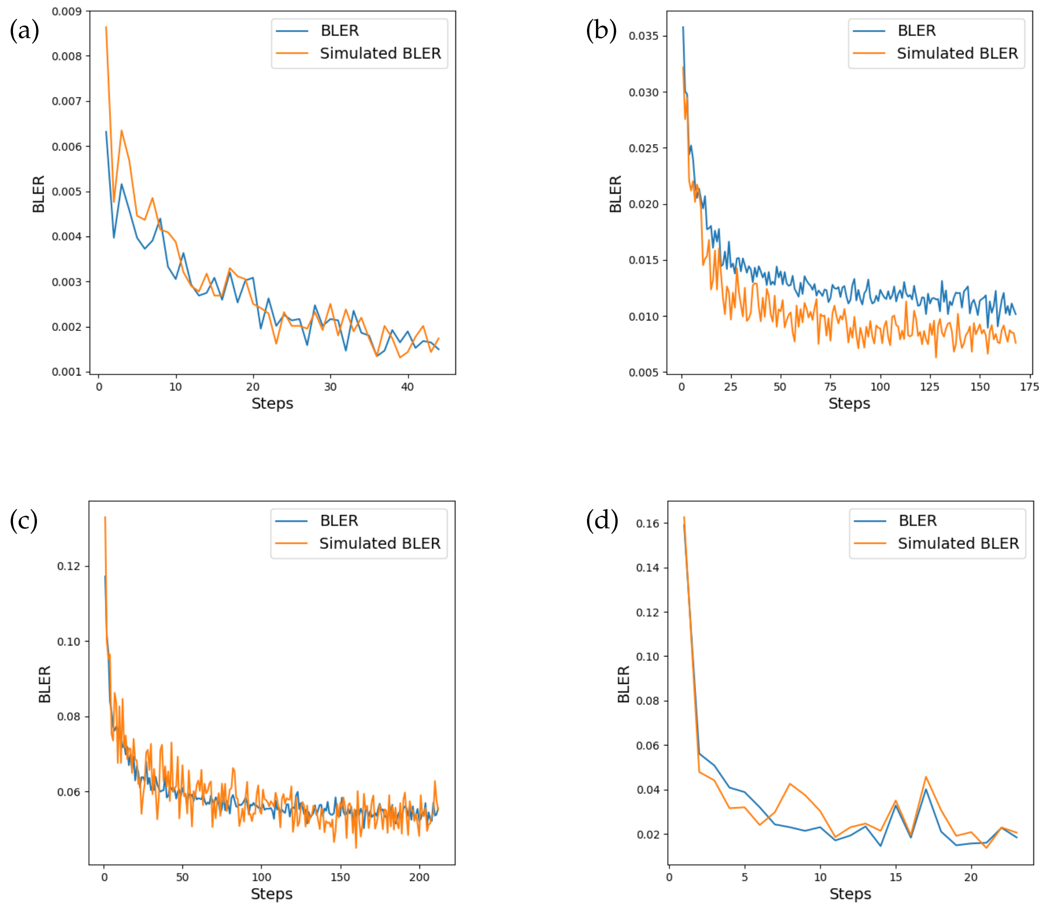


Figure 9. Comparison between BLER performance on the instantaneous channel transfer function and the simulated channel model recorded after each epoch of training. There is a high correlation between BLER between the true transfer function and the simulated channel model. The monitored BLER is shown in the (**a**) AWGN channel, (**b**) Rician fading channel, (**c**) Rayleigh fading channel, and (**d**) PA-AWGN channel.

In the field, evaluation of the true channel function may not be feasible after each epoch, hence monitoring performance will be reliant on the accuracy of the simulated channel model. If monitoring of the true channel performance is required, it is possible to intermittently deploy the transmitter weights to the origin side to evaluate performance at irregular intervals rather than every epoch. This is to decrease the frequency at which information-carrying transmissions are made during the training cycle and to maintain burst communications decreasing the chance of intercept.

Generative models provide a suitable method for enabling backpropagation in OAL, but the GAN method has been the subject of much research for learning in wireless communication systems without an assumed channel. Instead of concentrating on the GAN approach, we have instead proposed a simpler generative model capable of modelling the channel output distributions as shown in our results. By demonstrating equivalent performance to the joint end-to-end model, we are comparing our results to a model that has full knowledge of the simulated channel environment. In doing so, we demonstrate that the use of the MDN can provide a sufficient approximation of the true channel environment to permit the learning of an optimal code for that environment.

5. Limitations

As a generative model. the MDN is still vulnerable to mode collapse. Reasons for mode collapse in the MDN model are described in [33], suggesting that the primary reason is due to an imbalanced representation of data associated with modes in the training set. The authors suggest that the mixture components associated with the dominant modes represented in the data outweigh the other mixture coefficients and prevent variation in learning solutions [33]. They introduce additional loss terms that help to penalise large value weights and high variance parameters [33]. Our results did not suffer from mode collapse. One limitation of our approach is that we are using a small code size, and we are simulating a memoryless channel environment. In a real channel environment, certain channel states may persist for longer and therefore become over-represented during the training phase for the channel model. In a physical system, it may be more likely to encounter mode collapse and would require an exploration of approaches to mitigate this issue.

Additional limitations of our work include the assumption of ideal synchronisation. While the work focuses on the use of the MDN as a simpler alternative to generative modelling, scope remains for testing in more complex channels, which include timing, phase, and frequency offsets. These would lead to the additional considerations of a AE architecture suited to learning matched filtering and compensating for the additional channel effects. In keeping the experiments simple, we have also restricted our work to a short message length of $K = 8$ bits rather than investigating extension to longer codes.

The removal of the feedback channel has reduced the opportunity for data poisoning between the transmitter and receiver during training. However, there is still some potential for data poisoning during stage 2 of the learning procedure, when random uniform IQ symbols are transmitted over the true channel environment. This phase occurs as a regular burst transmission during each training iteration. While it may be possible to mitigate somewhat by reducing the regularity of this phase, the training of the channel model is reliant on sampling of uniform noise through the true channel environment.

6. Conclusions and Future Work

6.1. Conclusions

In this article, we have proposed an alternate generative channel model for training of transmitter and receiver OAL without relying on a feedback channel. We have shown that the MDN is able to model the distribution of a stochastic channel environment. As indicated by our results (Section 4), the proposed approach is able to produce an equivalent BLER to the joint end-to-end model, without a prior assumption of the channel function. We have demonstrated equivalent BLER performance for the $K = 8$ bit uncoded message in the AWGN, Rician fading, and Rayleigh fading channels as well as in the PA-AWGN channel. This is achieved without the need for a fixed channel model. Prior work has focused on the GAN model to learn the channel distribution, and while this has been shown to be effective, the training procedure does add complexity. We have demonstrated that the simpler MDN model, requiring only one Gaussian distribution per channel symbol, is a capable replacement for the GAN when modelling memoryless channels and does not require the overhead of a discriminator model during training. We have also shown that intermittent sampling of the end-to-end performance in terms of BLER is suitable for use as a stopping condition during the training process. This allows training to occur entirely on the receiver side without the need for feedback over the true channel, such as in RL based methods. The MDN is advantageous for OAL learning, does not require complex training regimes, and does not require multiple forward passes during inference (such as in the diffusion model). Removal of the feedback channel prevents the opportunity for data poisoning via feedback during the training process. The proposed approach is able to approximate the channel distribution such that the transmitter and receiver were able to learn an optimal code matching the performance of the joint end-to-end model.

6.2. Future Work

The limitations identified in Section 5 present opportunities for future work. Application to physical channel environments would necessarily require the addition of synchronisation. This would require modifications to both the transmitter and receiver architectures to support filtering and the ability to correct timing, phase, and frequency offsets in the receiver. To this end, it is possible to investigate extending the AE-based architecture to learn filtering, detection, and synchronisation. Conventional methods for synchronisation have been optimised for specific forms of filtering and modulation, yet it may be possible to combine data-driven and conventional approaches, such as in the work on deep unfolding [34]. In addition, physical channels may exhibit a certain degree of memory, as identified in [33], mode collapse in MDN is due to unequal representation of states during training. Future work will need to investigate the properties of channels with memory and investigate the effects on the MDN channel model as well as investigate methods to mitigate the unbalanced representation of states within the training data. While short codes have application in resource-constrained devices such as in the IoT, future work would also be required to make suitable modifications to the transmitter and receiver architectures to support longer codes or integrate with concatenated coding methods. Scalability due to message length and the challenges posed both to the architecture and to the sampling requirements of the learning process are areas that require further work for the practical application of DL methods to trainable wireless communications systems. Although the feedback path is no longer a vulnerability, there remains some potential for data poisoning the forward path during burst transmissions in stage 2 of training. Future work should investigate methods to mitigate this potential vulnerability by reducing the frequency of transmission, employing a low probability of detecting signalling as well as experimentally investigating robustness of the proposed method to data poisoning attacks. SDR platforms offer flexibility for defining novel wireless communications systems. The emergence of embedded and edge device hardware platforms supporting the optimisation required for the parallel computations required by DL will be necessary for translation of DL-based methods from software experimentation to hardware implementation. However, trainable wireless communication systems introduce a new paradigm where components of the system are no longer static and must support methods for redeployment and reconfiguration during operation.

Author Contributions: C.P.D.: Conceptualization; Data curation; Formal analysis; Investigation; Methodology; Project administration; Resources; Software; Validation; Visualization; Writing—original draft; Writing—review and editing. I.S.: Conceptualization; Resources; Project administration; Supervision; Writing—review and editing. R.C.D.: Resources; Project administration; Supervision; Writing—review and editing. S.S.-S.: Writing—review and editing; Resources. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by UniSQ-DSTG Postgraduate Research Scholarship 2021-2024 on the 'Design of Efficient Artificial Intelligence Algorithms for Future Communication Systems'. It is funded by the Department of Defence, Commonwealth of Australia under DSP Scholarship (Project-Based) Agreement 10254.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

The following abbreviations are used in this manuscript:

Adam	adaptive moment estimation
AE	autoencoder

AWGN	additive white Gaussian noise
BLER	block error rate
$BLER_{sim}$	simulated channel block error rate
BPSK	binary phase shift keying
CE	cross-entropy
CNN	convolutional neural network
DDIM	denoising diffusion implicit model
DDPM	diffusion-denoising probabilistic model
DL	deep learning
GAN	generative adversarial network
IoT	internet of things
IQ	in-phase and quadrature
MDN	mixture density network
NLL	negative log-likelihood
OAL	over-the-air learning
PA-AWGN	power amplifier with an additive white Gaussian noise
QPSK	quadrature phase-shift keying
RA-GAN	residual aided generative adversarial network
ReLU	rectified linear unit
RL	reinforcement learning
SDR	software-defined radio
SISO	single input single output
SNR	signal to noise ratio
SPSA	simultaneous perturbation stochastic approximation
SSPA	solid-state high-power amplifier
SWA	stochastic weight averaging
WGAN	Wasserstein generative adversarial network

References

1. O'Shea, T.; Hoydis, J. An Introduction to Deep Learning for the Physical Layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [CrossRef]
2. Raj, V.; Kalyani, S. Backpropagating through the Air: Deep Learning at Physical Layer without Channel Models. *IEEE Commun. Lett.* **2018**, *22*, 2278–2281. [CrossRef]
3. Aoudia, F.A.; Hoydis, J. End-to-End Learning of Communications Systems without a Channel Model. In Proceedings of the 2018 52nd Asilomar Conference on Signals, Systems, and Computers, Grove, CA, USA, 28–31 October 2018; pp. 298–303. [CrossRef]
4. Ye, H.; Li, G.Y.; Juang, B.F.; Sivanesan, K. Channel Agnostic End-to-End Learning Based Communication Systems with Conditional GAN. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–5. [CrossRef]
5. O'Shea, T.J.; Roy, T.; West, N. Approximating the Void: Learning Stochastic Channel Models from Observation with Variational Generative Adversarial Networks. In Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019; pp. 681–686. [CrossRef]
6. Sagduyu, Y.E.; Shi, Y.; Erpek, T. Adversarial Deep Learning for Over-the-Air Spectrum Poisoning Attacks. *IEEE Trans. Mob. Comput.* **2021**, *20*, 306–319. [CrossRef]
7. Dörner, S.; Henninger, M.; Cammerer, S.; Brink, S.T. WGAN-based Autoencoder Training Over-the-air. In Proceedings of the 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Atlanta, GA, USA, 26–29 May 2020; pp. 1–5. [CrossRef]
8. Davey, C.P.; Shakeel, I.; Deo, R.C.; Salcedo-Sanz, S. Channel-Agnostic Training of Transmitter and Receiver for Wireless Communications. *Sensors* **2023**, *23*, 9848. [CrossRef] [PubMed]
9. Dörner, S.; Cammerer, S.; Hoydis, J.; Brink, S.T. Deep Learning Based Communication Over the Air. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 132–143. [CrossRef]
10. Cammerer, S.; Aoudia, F.A.; Dörner, S.; Stark, M.; Hoydis, J.; Brink, S.T. Trainable Communication Systems: Concepts and Prototype. *IEEE Trans. Commun.* **2020**, *68*, 5489–5503. [CrossRef]
11. Ye, H.; Liang, L.; Li, G.Y.; Juang, B.H. Deep Learning-Based End-to-End Wireless Communication Systems with Conditional GANs as Unknown Channels. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3133–3143. [CrossRef]
12. Alawad, M.A.; Hamdan, M.Q.; Hamdi, K.A. Innovative Variational AutoEncoder for an End-to-end Communication System. *IEEE Access* **2022**, *11*, 86834–86847. [CrossRef]
13. Jiang, H.; Bi, S.; Dai, L.; Wang, H.; Zhang, J. Residual-Aided End-to-End Learning of Communication System without Known Channel. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 631–641. [CrossRef]

14. Kim, M.; Fritschek, R.; Schaefer, R.F. Learning End-to-End Channel Coding with Diffusion Models. In Proceedings of the WSA & SCC 2023; 26th International ITG Workshop on Smart Antennas and 13th Conference on Systems, Communications, and Coding, Braunschweig, Germany, 27 February 2023; VDE: Frankfurt am Main, Germany, 2023; pp. 1–6. [CrossRef]
15. Aoudia, F.A.; Hoydis, J. Model-free training of end-to-end communication systems. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2503–2516. [CrossRef]
16. Goodfellow, I. Nips 2016 tutorial: Generative adversarial networks. *arXiv* **2016**, arXiv:1701.00160. [CrossRef]
17. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, NSW, Australia, 6–11 August 2017; pp. 214–223. [CrossRef]
18. Bishop, C.M. Mixture Density Networks. Technical Report, Aston University. 1994. Available online: https://research.aston.ac.uk/en/publications/mixture-density-networks (accessed on 10 October 2023).
19. Unni, R.; Yao, K.; Zheng, Y. Deep Convolutional Mixture Density Network for Inverse Design of Layered Photonic Structures. *ACS Photonics* **2020**, *7*, 2703–2712. [CrossRef] [PubMed]
20. Karoliny, J.; Etzlinger, B.; Springer, A. Mixture Density Networks for WSN Localization. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 7–11 June 2020; pp. 1–5. [CrossRef]
21. Mostafavi, S.; Sharma, G.P.; Gross, J. Data-Driven Latency Probability Prediction for Wireless Networks: Focusing on Tail Probabilities. In Proceedings of the GLOBECOM 2023–2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, 4–8 December 2023; pp. 4338–4344. [CrossRef]
22. Khurjekar, I.D.; Gerstoft, P. Multi-Source DOA Estimation With Statistical Coverage Guarantees. In Proceedings of the ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 5310–5314. [CrossRef]
23. Li, H.; Kanuric, T.; Eichberger, A. Automotive Radar Modeling for Virtual Simulation Based on Mixture Density Network. *IEEE Sensors J.* **2023**, *23*, 11117–11124. [CrossRef]
24. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 448–456. [CrossRef]
25. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941. [CrossRef]
26. Veit, A.; Wilber, M.J.; Belongie, S. Residual networks behave like ensembles of relatively shallow networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 550–558. [CrossRef]
27. Dubey, S.R.; Singh, S.K.; Chaudhuri, B.B. A Comprehensive Survey and Performance Analysis of Activation Functions in Deep Learning. *Neurocomputing* **2022**, *503*, 92–108. [CrossRef]
28. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323. [CrossRef]
29. Rapp, C. Effects of HPA-nonlinearity on a 4-DPSK/OFDM-signal for a digital sound broadcasting signal. *ESA Spec. Publ.* **1991**, *332*, 179–184. [CrossRef]
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]
31. Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; Wilson, A.G. Averaging weights leads to wider optima and better generalization. *arXiv* **2018**, arXiv:1803.05407. [CrossRef]
32. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017, pp. 464–472. [CrossRef]
33. Zhou, Y.; Gao, J.; Asfour, T. Movement primitive learning and generalization: Using mixture density networks. *IEEE Robot. Autom. Mag.* **2020**, *27*, 22–32. [CrossRef]
34. Jagannath, A.; Jagannath, J.; Melodia, T. Redefining Wireless Communication for 6G: Signal Processing Meets Deep Learning With Deep Unfolding. *IEEE Trans. Artif. Intell.* **2021**, *2*, 528–536. [CrossRef]

7.3 Links and Implications

The article proposed a method of OAL for transmitter, channel and receiver which did not require feedback. In addition, the proposed MDN channel model does not require multiple models or iterations during training and inference to approximate the channel distribution. Both features provide further efficiencies and simplification for OAL as opposed to the methods presented in literature. In addition, the article considers adversarial channel environments by training the channel with randomised values rather than an information carrying signal. The article demonstrated that the resulting model is able to approximate a variety of channel distributions even though it did not train on the resulting modulations. Furthermore, the article considers a practical issue of measuring the system performance during training in OAL systems, which will not necessarily access to the true channel during training. By using the BLER of the system produced through the developed channel model, the article shows that this metric is strongly correlated with the BLER produced over the true channel during training and can therefore be used as a suitable stopping condition. Training of DL based wireless communications systems OAL is an important capability which enables optimisation to channel environments which are otherwise difficult to simulate. Such methods are made more achievable by providing more efficient methods to channel modelling and simplifying approaches to training algorithms, and more secure by limiting the opportunity for adversarial attacks during training.

CHAPTER 8: CONCLUSION AND FUTURE SCOPE

8.1 Summary

The design and development of wireless communications systems faces new challenges for data-transfer rates, reliability, availability, privacy and security driven by the evolution of applications which rely on mobility, interconnection and the large scale collection and sharing of information. At the same time, we have seen the broader application of artificial intelligence-based techniques which have the capabilities to learn from observation and perform global optimisation in a variety of engineering domains. DL is a framework which exhibits the ability to fuse a variety of high dimensional data sources to extract useful representations with the aim of task driven optimisation. As such, it has drawn attention of researchers in wireless communications systems as a potential design tool which can be used to help address some of the key challenges emerging in the field.

The work undertaken in this doctoral thesis has focused on three domains relating to the use of DL in the design of wireless communications systems; Synchronisation, Adaptation and Over-the-air Learning (OAL).

Synchronisation (achieved through Objective 1) is a signal processing task that corrects perturbations in the received signal for the accurate recovery of the transmitted message at the receiver. DL based estimators have been applied to various synchronisation parameters. However, due to the wide application of parameter estimation in waveforms, such as for frequency offset estimation, these methods have often been applied in ways that are isolated from the communications system. When DL has been applied in synchronisation for communications systems, prior work has not considered the impact of the type of modulation, or structure of preamble on the model accuracy. In Chapter 4, this thesis has developed a novel DL model that performs multi-stage estimation for CFO on shortened preambles. Short preambles are advantageous where data-rate is concerned since by reducing the preamble length, the amount of allocated channel usage for information symbols and consequently data-rate can be maximised. We consider multiple modulations in the communications setting, and include Chaotic map functions, as these provide potential advantages for physical layer security with application to the IoT. The accuracy of conventional FFT based CFO estimation methods are dependent on the preamble length and SNR, and this work demonstrates that a DL based approach achieves better accuracy under both of these constraints. Similarly, correlation-based methods are dependent on the number of samples available, and the proposed method demonstrates greater accuracy than correlation-based CFO estimation with limited upsampling. The branching DL architecture is a novel approach capable of refining the CFO estimate in two stages, and the advantages of estimating the error term using regression are clearly demonstrated. Further, conventional signal processing algorithms focus on certain features of the incoming signal, the research in this thesis has shown that incorporating feature processing derived from these algorithms improves the accuracy of the resulting DL model. Importantly, to support the security of wireless communications and avoid the potential for adversarial playback attacks, it is important for a synchronisation method to be able to support randomised as opposed to fixed preambles. The work in this thesis demonstrates that DL is able to outperform conventional CFO estimation methods on both fixed and random sequences, and that a single model can be applied on multiple modulations. This work also demonstrates that the accuracy of CFO estimation for both conventional and data-driven methods are impacted by the structure and type of modulation. This result is particularly interesting, as it indicates that the choice of preamble design, and the properties of the preamble modulation are impactful for the overall performance of the system.

Adaptation (achieved through Objective 2) in wireless communications systems changes the transmitted modulation and coding in response to measurements of changing channel conditions. In particular AMC schemes are shown to achieve higher reliability under changing channel conditions than a single coded modulation alone. However, E2E learning methods for DL described in literature support only a single coded modulation, requiring multiple AE to be trained to realise multiple schemes. In Chapter 5, a novel DL architecture and custom training algorithm is proposed for AMC framed as a multi-task learning problem. This is a significant extension of the E2E learning framework to support AMC schemes. The resulting method is demonstrated to achieve gains over conventional codes in two Rayleigh fading channels while only being trained on an AWGN channel. Consequently demonstrating the ability to a DL based AMC to provide spectral efficiency under changing channel conditions without the requirement for retraining. This capability contributes to the reliability and availability for future wireless communications systems under a variable channel environment.

Over-the-air learning (OAL) (achieved through Objectives 3 & 4) is an extension of E2E learning algorithms to support optimisation of DL wireless communications systems over an unknown channel, alleviating the requirement for an assumed channel model. This is advantageous as the DL based transmitter and receiver can be optimised for communicating over channel environments that may not be easily simulated or have a analytic form. Alternately, tuning in OAL supports the adaptation of pre-trained wireless communications systems to varying channel conditions (as an alternative or complementary technique for AMC). Current approaches to training in OAL leverage techniques such as RL to perform gradient approximation, use GAN for channel approximation or leverage coordination protocols for cooperative agent learning. However, each of these methods have disadvantages, both in terms of complexity, channel usage and security during training.

Chapter 6 proposes a simplification over these methods by developing a novel training algorithm which learns implicit channel information through the estimates passed between a remote and local receiver. The article demonstrates that the local receiver learns enough implicit information about the channel environment to optimise the transmitter during the joint learning process. This alleviates the need for a separate channel model and does not require multiple forward passes during the training process to approximate gradients from the receiver. It is demonstrated to provide equivalent performance to the original E2E method and produce gains over conventional codes in a variety of channel environments.

In Chapter 7, the feedback channel is removed, and instead a MDN is trained to perform channel approximation to enable E2E training on the receiver. The removal of the feedback channel enhances the security aspect of OAL training by reducing the potential for adversarial detection and replay attacks. In addition, the proposed approach does not use an information carrying signal to train the channel model, instead it transmits a uniform random sequence to generate the training batch, and this is shown to enable the resulting channel model to approximate the true channel distribution for both conventional BPSK and the resulting learnt modulations. The lack of an information carrying signal, with regular repeating symbols, contributes to the security of OAL training by reducing the likelihood of detection by an eavesdropper and removes dependency on structured communication to train the channel model. In addition, the use of the MDN to model the channel distribution simplifies existing approaches by removing the need for additional models such as the discriminator in GAN training and removing the need for iterative training such as is used in diffusion type models. By promoting simpler models and more secure training algorithms for OAL the research contributes to the reliability, availability and security of future wireless communications systems.

8.2 Limitations and Directions for Future Research

The work undertaken in this doctoral thesis has applied a number of constraints in each area to examine the proposed methods in each featured publication.

In addressing **Synchronisation** this thesis focused on CFO which in literature has been demonstrated to be more difficult to obtain an accurate estimator. Parameter estimation for timing offsets and detection was not addressed. In addition, the resulting model was examined as a parameter estimator separate from an integrated wireless communications system, in order to evaluate the accuracy of the CFO estimation. In future research there is the potential to examine integration of DL based synchronisation as part of an E2E wireless communications system, both within a conventional and a DL based design. The publication focused on parameter estimation as the objective task for the developed DL model, however, in an integrated system, the resulting message recovery in BER and BLER would serve as the main objective. Therefore, in future work, integrated DL based synchronisation may focus on correcting the perturbations of the received signal rather than estimation of a set of parameters which may be applied independently.

The proposed DL architecture for **Adaptation** via AMC consisted of a discrete gate which was parameterised to allow learning for different codes. In this sense the switching mechanism between different code rates was not learned as part of the network, instead it was a predefined conditional operator that was built into the network. Instead, future work could devise a mechanism where the discrete gate could instead be learnt, however this is challenging because it would require learning how to disable other pathways through the network and may not be as efficient as the proposed method, due to redundant activations among different network branches. Future work would investigate tree structures in DL, which may be connected to graph neural network architectures. The research in this thesis did not explore the learning of mapping between channel conditions such as SNR and optimal code rate. The existing literature has focused more on this task as a parameter estimation or classification problem or used RL to learn a control algorithm for existing coded modulation. Instead, future work could investigate whether it is possible to incorporate several stages in learning, those which extract features from the channel environment, the learning of coded modulation schemes, and the learning of an optimal control policy. This type of work would investigate how such a training algorithm may be devised, whether it is possible to train such a system E2E, iteratively or in separate stages, and the practicality of each training regime. As mentioned in the article, the industrial internet of things (IIoT) and wireless sensor networks (WSNs) are subject to additional constraints such as energy use. There is potential for future work to consider these other constraints in the learning objective for AMC to enhance the availability of these type of systems.

Proposed methods for **Over-the-air learning (OAL)** were developed under the simplifying assumptions of perfect synchronisation and for single-input single-output (SISO) wireless communications systems. As mentioned, future research would consider the integration of learnt synchronisation methods for additional perturbations of frequency, phase and timing offsets. In addition, there is opportunity to consider MIMO wireless communication systems which must handle multiple carriers and expands the dimensionality of the system model. Existing E2E DL based communications systems methods are limited by the dimensionality of the message, especially where a symbol-wise representation is used. However, as the message increases in dimension 2^K it becomes impractical to produce such extremely large classifiers not only in terms of dimension but also due to sampling problems from the input message domain during data generation (it becomes impossible to sample all possible messages). Bit-wise approaches have investigated training E2E AE on individual message bits, and various architectures have been proposed. In addition, the use of small codes as an inner concatenated code with an outer LDPC code has been demonstrated to achieve much longer message lengths. However, future work may also investigate structuring messages as multiple blocks, as well as the use of CNN and RNN in learning variable length message sizes. Security is also a primary concern for wireless communications systems, as due to the channel environment, they are exposed to eavesdropping and adversarial attacks such as replay attacks. The development of LPI and low probability of detect (LPD) codes reduces the probability of messages being understood or detected by an eavesdropper. This results in covert communications schemes. While the codes learnt by the AE models leverage a more continuous IQ space than conventional codes, there is future scope to investigate the E2E learning of LPI and LPD codes through application of appropriate constraints on the DL model, such as through limiting peak-to-average power ratio (PAPR) of learnt codes. Finally, the methods developed in this thesis were reliant on simulation of the channel environment. Future work will have the challenge of implementing training regimes on physical hardware such as SDR over a physical channel environment. However, there may be limited practical opportunity to test and train such systems in a variety of complex channels such as selective-fading channels or especially channels relating to space communications. Instead, there remains scope for the simulation of more sophisticated channels to train, evaluate and develop the proposed and new methods in future research.

References

[1] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjoland, F. Tufvesson, 6g wireless systems: Vision, requirements, challenges, insights, and opportunities, Proceedings of the IEEE 109 (7) (2021) 1166–1199. doi:10.1109/JPROC.2021.3061701.

[2] W. Jiang, B. Han, M. A. Habibi, H. D. Schotten, The road towards 6g: A comprehensive survey, IEEE Open Journal of the Communications Society 2 (2021) 334–366. doi:10.1109/OJCOMS.2021.3057679.

[3] T. O'Shea, J. Hoydis, An introduction to deep learning for the physical layer, IEEE transactions on cognitive communications and networking 3 (4) (2017) 563–575. doi:10.1109/tccn.2017.2758370.

[4] Y. Bengio, Y. Lecun, G. Hinton, Deep learning for ai, Commun. ACM 64 (7) (2021) 58–65. doi:10.1145/3448250.

[5] Z. Qin, H. Ye, G. Y. Li, B.-H. F. Juang, Deep learning in physical layer communications, IEEE wireless communications 26 (2) (2019) 93–99. doi:10.1109/MWC.2019.1800601.

[6] C. Mao, Z. Mu, Q. Liang, I. Schizas, C. Pan, Deep learning in physical layer communications: Evolution and prospects in 5g and 6g networks, IET Communications 17 (2023) 1863–1876. doi:10.1049/cmu2.12669.

[7] W. G. Cowley, Phase and frequency estimation for psk packets: bounds and algorithms, IEEE transactions on communications 44 (1) (1996) 26–28. doi:10.1109/26.476092.

[8] J. Zhang, T. Duong, R. Woods, A. Marshall, Securing wireless communications of the internet of things from the physical layer, an overview, Entropy (Basel, Switzerland) 19 (8) (2017) 420. doi:10.3390/e19080420.

[9] A. J. Michaels, Digital chaotic communications, Doctoral dissertation, Georgia Institute of Technology, Georgia, United States (2009).
URL http://hdl.handle.net/1853/34849

[10] B. Jovic, Synchronization techniques for chaotic communication systems, Springer Science & Business Media, Berlin, 2011. doi:10.1007/978-3-642-21849-1.

[11] J. Tao, L. Yang, X. Han, Enhanced carrier frequency offset estimation based on zadoff-chu sequences, IEEE communications letters 23 (10) (2019) 1862–1865. doi:10.1109/LCOMM.2019.2930700.

[12] F. A. Aoudia, J. Hoydis, End-to-end learning of communications systems without a channel model, in: 2018 52nd Asilomar Conference on Signals, Systems, and Computers, 2018, pp. 298–303. doi:10.1109/ACSSC.2018.8645416.

[13] H. Ye, G. Y. Li, B. F. Juang, K. Sivanesan, Channel agnostic end-to-end learning based communication systems with conditional gan, in: 2018 IEEE Globecom Workshops (GC Wkshps), 2018, pp. 1–5. doi:10.1109/GLOCOMW.2018.8644250.

[14] J. Downey, D. Mortensen, M. Evans, J. Briones, N. Tollis, Adaptive coding and modulation experiment using nasa's space communication and navigation testbed, in: 2016 Communications Satellite Systems Conference (ICSSC), 2016. doi:10.2514/6.2016-5736.

[15] I. Hwang, T. Jang, M. Kang, S. No, J. Son, D. Hong, C. Kang, Performance analysis of adaptive modulation and coding combined with transmit diversity in next generation mobile communication systems, Future generation computer systems 20 (2) (2004) 189–196. doi:10.1016/S0167-739X(03)00133-X.

[16] G. Caire, K. R. Kumar, Information theoretic foundations of adaptive coded modulation, Proceedings of the IEEE 95 (12) (2007) 2274–2298. doi:10.1109/JPROC.2007.904444.

[17] M. Ibnkahla, Applications of neural networks to digital communications – a survey, Signal processing 80 (7) (2000) 1185–1215. doi:10.1016/S0165-1684(00)00030-X.

[18] H. Ye, G. Y. Li, B.-H. Juang, Deep learning based end-to-end wireless communication systems without pilots, IEEE transactions on cognitive communications and networking 7 (3) (2021) 702–714. doi:10.1109/TCCN.2021.3061464.

[19] H. Ye, L. Liang, G. Y. Li, B.-H. Juang, Deep learning-based end-to-end wireless communication systems with conditional gans as unknown channels, IEEE transactions on wireless communications 19 (5) (2020) 3133–3143. doi:10.1109/TWC.2020.2970707.

[20] T. J. O. Shea, K. Karra, T. C. Clancy, Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention, in: 2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2016, pp. 223–228. doi:10.1109/ISSPIT.2016.7886039.

[21] V. Ninkovic, A. Valka, D. Dumic, D. Vukobratovic, Deep learning based packet detection and carrier frequency offset estimation in ieee 802.11ah, IEEE Access. 9 (2021) 99853–99865. doi:10.1109/ACCESS.2021.3096853.

[22] R. M. Dreifuerst, R. W. Heath, M. N. Kulkarni, J. Charlie, Deep learning-based carrier frequency offset estimation with one-bit adcs, in: 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2020, pp. 1–5. doi:10.1109/SPAWC48557.2020.9154214.

[23] E.-R. Jeong, E.-S. Lee, J. Joung, H. Oh, Convolutional neural network (cnn)-based frame synchronization method, Applied Sciences 10 (20) (2020) 7267. doi:10.3390/app10207267.

[24] I. Sajedian, J. Rho, Accurate and instant frequency estimation from noisy sinusoidal waves by deep learning, Nano Convergence 6 (1) (2019) 27. doi:10.1186/s40580-019-0197-y.

[25] G. Izacard, B. Bernstein, C. Fernandez-Granda, A learning-based framework for line-spectra super-resolution, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 3632–3636. doi:10.1109/ICASSP.2019.8682882.

[26] R. H. Barker, Group synchronization of binary digital system, Communication theory (1953) 273–287.

[27] A. Berni, W. Gregg, On the utility of chirp modulation for digital signaling, IEEE Transactions on Communications 21 (6) (1973) 748–751. doi:10.1109/TCOM.1973.1091721.

[28] B. M. Popovic, Generalized chirp-like polyphase sequences with optimum correlation properties, IEEE Transactions on Information Theory 38 (4) (1992) 1406–1409. doi:10.1109/18.144727.

[29] S. Kalade, L. H. Crockett, R. W. Stewart, Training deep filters for physical-layer frame synchronization, IEEE Open Journal of the Communications Society 3 (2022) 1063–1075. doi:10.1109/OJCOMS.2022.3185973.

[30] G. Schreiber, M. Tavares, 5g new radio physical random access preamble design, in: 2018 IEEE 5G World Forum (5GWF), 2018, pp. 215–220. doi:10.1109/5GWF.2018.8517052.

[31] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science (American Association for the Advancement of Science) 313 (5786) (2006) 504–507. doi:10.1126/science.1127647.

[32] S. Cammerer, F. A. Aoudia, S. Dörner, M. Stark, J. Hoydis, S. t. Brink, Trainable communication systems: Concepts and prototype, IEEE Transactions on Communications 68 (9) (2020) 5489–5503. doi:10.1109/TCOMM.2020.3002915.

[33] F. A. Aoudia, J. Hoydis, Model-free training of end-to-end communication systems, IEEE Journal on Selected Areas in Communications 37 (11) (2019) 2503–2516. doi:10.1109/JSAC.2019.2933891.

[34] T. J. O'Shea, K. Karra, T. C. Clancy, Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention, in: 2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2016, pp. 223–228. doi:10.1109/ISSPIT.2016.7886039.

[35] T. J. O'Shea, L. Pemula, D. Batra, T. C. Clancy, Radio transformer networks: Attention models for learning to synchronize in wireless systems, 2016 50th Asilomar Conference on Signals, Systems and Computers (2016) 662 – 666doi:10.1109/ACSSC.2016.7869126.

[36] J. Schmitz, C. v. Lengerke, N. Airee, A. Behboodi, R. Mathar, A deep learning wireless transceiver with fully learned modulation and synchronization, in: 2019 IEEE International Conference on Communications Workshops (ICC Workshops), 2019, pp. 1–6. doi:10.1109/ICCW.2019.8757051.

[37] S. Dörner, S. Cammerer, J. Hoydis, S. t. Brink, Deep learning based communication over the air, IEEE Journal of Selected Topics in Signal Processing 12 (1) (2018) 132–143. doi:10.1109/JSTSP.2017.2784180.

[38] V. Raj, S. Kalyani, Backpropagating through the air: Deep learning at physical layer without channel models, IEEE Communications Letters 22 (11) (2018) 2278–2281. doi:10.1109/LCOMM.2018.2868103.

[39] R. S. Sutton, D. McAllester, S. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: S. Solla, T. Leen, K. Müller (Eds.), Advances in Neural Information Processing Systems, Vol. 12, MIT Press, 1999.
URL https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf

[40] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint arXiv:1509.02971 (2015). doi:10.48550/arXiv.1509.02971.

[41] B. Zhang, N. Van Huynh, Deep deterministic policy gradient for end-to-end communication systems without prior channel knowledge, arXiv preprint arXiv:2305.07448 (2023). doi:10.48550/arXiv.2305.07448.

[42] T. J. O'Shea, T. Roy, N. West, B. C. Hilburn, Physical layer communications system design over-the-air using adversarial networks, in: 2018 26th European Signal Processing Conference (EUSIPCO), 2018, pp. 529–532. doi:10.23919/EUSIPCO.2018.8553233.

[43] T. J. O'Shea, T. Roy, N. West, Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks, in: 2019 International Conference on Computing, Networking and Communications (ICNC), 2019, pp. 681–686. doi:10.1109/ICCNC.2019.8685573.

[44] S. Dörner, M. Henninger, S. Cammerer, S. t. Brink, Wgan-based autoencoder training over-the-air, in: 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2020, pp. 1–5. doi:10.1109/SPAWC48557.2020.9154335.

[45] I. Goodfellow, Nips 2016 tutorial: Generative adversarial networks, arXiv preprint arXiv:1701.00160 (2016). doi:10.48550/arXiv.1701.00160.

[46] M. Kim, R. Fritschek, R. F. Schaefer, Learning end-to-end channel coding with diffusion models, in: WSA & SCC 2023; 26th International ITG Workshop on Smart Antennas and 13th Conference on Systems, Communications, and Coding, VDE, 2023, pp. 1–6. URL https://ieeexplore.ieee.org/document/10104549

[47] C. de Vrieze, S. Barratt, D. Tsai, A. Sahai, Cooperative multi-agent reinforcement learning for low-level wireless communication, arXiv preprint arXiv:1801.04541 (2018). doi:10.48550/arXiv.1801.04541.

[48] A. Sahai, J. Sanz, V. Subramanian, C. Tran, K. Vodrahalli, Learning to communicate with limited co-design, in: 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2019, pp. 184–191. doi:10.1109/ALLERTON.2019.8919749.

[49] A. Adeel, H. Larijani, A. Ahmadinia, Random neural network based cognitive engines for adaptive modulation and coding in lte downlink systems, Computers & electrical engineering 57 (2017) 336–350. doi:10.1016/j.compeleceng.2016.11.005.

[50] S. Kojima, K. Maruta, C. J. Ahn, Adaptive modulation and coding using neural network based snr estimation, IEEE Access 7 (2019) 183545–183553. doi:10.1109/ACCESS.2019.2946973.

[51] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilen, R. C. Reinhart, D. J. Mortensen, Reinforcement learning for satellite communications: From leo to deep space operations, IEEE Communications Magazine 57 (5) (2019) 70–75. doi:10.1109/MCOM.2019.1800796.

[52] H. Safi, M. Akbari, E. Vaezpour, S. Parsaeefard, R. M. Shubair, Autoencoder-bank based design for adaptive channel-blind robust transmission, EURASIP journal on wireless communications and networking 2021 (1) (2021) 1–15. doi:10.1186/s13638-021-01929-z.

[53] D. Adams, The Restaurant at the End of the Universe: Hitchhiker's Guide to the Galaxy Book 2, Vol. 2, Tor UK, 1989.

[54] J. Proakis, M. Salehi, Digital communications, 5th Edition, McGraw-Hill, New York, 2008.

[55] C. Rapp, Effects of hpa-nonlinearity on a 4-dpsk/ofdm-signal for a digital sound broadcasting signal, ESA Special Publication 332 (1991) 179–184.
URL `https://ui.adsabs.harvard.edu/abs/1991ESASP.332..179R`

Though no one of us will ever be able to step back far enough to see the "big picture" we shouldn't forget that it exists. We should remember that physical law is what makes it all happen - way, way down in neural nooks and crannies which are too remote for us to reach with our high-level introspective probes.

Douglas R. Hofstadter,
Gödel, Escher, Bach:An Eternal Golden Braid.
New York: Basic Books 1979. p710