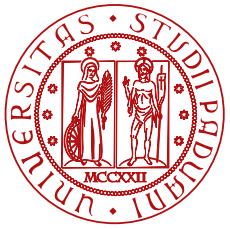

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in ICT for Internet and Multimedia

Joint Source-Channel Coding for Two-Hop Channels in Multi-hop Semantic Communication: Image Transmission Study

Master Candidate

Mohsen Kordi

Student ID 2040503

Supervisor

Prof. Federico Chiariotti

University of Padova

Academic Year
2022/2023
30 November 2023

This thesis is dedicated to my mother and father, my sister, and my brothers, whose love and unwavering belief in my abilities have been a guiding light. Your support has been the cornerstone of my academic achievements, and I am forever grateful for your encouragement and sacrifices.

Abstract

This research examines two models, Decoding and Re-encoding (DREC) and Joint Coding With Re-amplification (JCWR), specifically designed for two-hop wireless communication networks. The main goal is to investigate challenges caused by noise and interference in multi-hop communication scenarios. DREC uses two autoencoders and two communication channels, making it adaptable to different channel conditions and allowing for flexible resource utilization. In contrast, JCWR uses a single autoencoder with an amplifier, making it simpler and more adaptable for resource allocation. The study includes a thorough examination of these models' performance under various conditions, such as different compression ratios and signal-to-noise ratios, using the COCO 2017 image dataset. Additionally, the study explores extending both models to multi-link scenarios, providing insights into their scalability and performance. Future research directions include enhancing security measures, addressing latency issues, and investigating advanced amplification techniques.

Sommario

Questa ricerca esamina due modelli, DREC e JCWR, specificatamente progettati per reti di comunicazione wireless a due salti. L'obiettivo principale è indagare le sfide causate dal rumore e dalle interferenze negli scenari di comunicazione multi-hop. DREC utilizza due codificatori automatici e due canali di comunicazione, rendendolo adattabile alle diverse condizioni del canale e consentendo un utilizzo flessibile delle risorse. Al contrario, JCWR utilizza un singolo autocodificatore con un amplificatore, rendendolo più semplice e più adattabile per l'allocazione delle risorse. Lo studio include un esame approfondito delle prestazioni di questi modelli in varie condizioni, come diversi rapporti di compressione e rapporti segnale-rumore, utilizzando il set di dati di immagini COCO 2017. Inoltre, lo studio esplora lestensione di entrambi i modelli a scenari multi-link, fornendo approfondimenti sulla loro scalabilità e prestazioni. Le direzioni future della ricerca includono il miglioramento delle misure di sicurezza, la risoluzione dei problemi di latenza e lo studio di tecniche di amplificazione avanzate.

Contents

List of Figures

List of Acronyms

ADJSCC Attention DL-based JSCC

ANN Artificial Neural Network

AUC Area Under Curve

AWGN Additive White Gaussian Noise

CAE Convolutional Autoencoder

CNN Convolutional Neural Networks

ConvLSTM Convolutional Long Short-Term Memory

CR Compression Ratio

CSIR Channel State Information at the Receiver

CSNR Channel SNR

DNN Deep Neural Networks

DL Deep Learning

DREC Decoding and Re-encoding

GAN Generative Adversarial Network

GRU Gated Recurrent Unit

I.I.D. Independent Identically Distributed

IoT Internet of Things

JCWM Joint Coding With Mapping

JCWR Joint Coding With Re-amplification

JSCC Joint Source-Channel Coding

LSTM Long Short-Term Memory

MIMO Multiple Input, Multiple Output

MSE Mean Square Error

MISO Multiple Input, Single Output

ML Machine Learning

MOS Mean Opinion Score

NN Neural Network

PCA Principal Component Analysis

PReLU Parametric Rectified Linear Unit

PSNR Peak Signal-to-Noise Ratio

RNN Recurrent Neural Network

SIMO Single Input, Multiple Output

SNR Signal to Noise Ratio

SSIM Structural Similarity Index

1

Introduction

Communication systems are essential for sending information across different means, making it possible for people, businesses, and the world at large to share ideas, data, and multimedia content. These systems involve a sender, who creates the message, and a receiver, who understands it. To make sure the message gets through correctly and quickly, communication systems use source coding and channel coding techniques.

Source encoding, also known as data compression, changes the original message into a smaller form. This process tries to make the message smaller while keeping all the important information. The main goal of source encoding is to get rid of the extra information in the message so that it can be sent and stored more easily. Source encoding algorithms use the patterns and statistics in the data to compress it. For example, in normal language, some letters or words are more common than others. Source encoding algorithms can use these patterns to make the message smaller.

Channel encoding, also known as error-control coding, is a technique used to protect data from being corrupted during transmission. It works by adding redundant information to the data, which allows the receiver to detect and correct errors that occur during the transmission process. Data can be corrupted for various reasons, including noise, interference, and other transmission errors. Noise refers to random disturbances in the transmission channel that can

alter the data, while interference refers to external signals that can overlap with the data and cause distortion.

Claude Shannon, a pioneer in information theory, established the separation theorem, a fundamental result in communication theory. This theorem states that, under certain conditions, source encoding and channel encoding can be performed independently without sacrificing the optimality of the rate. In essence, Shannon's separation theorem implies that it is possible to design separate source and channel codes that achieve the best possible performance in terms of error rate and compression efficiency. This separation simplifies the design process and allows for modularity in communication system design [11].

However, the optimality of separation in Shannons theorem assumes no limitations on the complexity of source and channel code design and is based on idealized conditions, such as the use of an AWGN channel and perfect source coding [11]. In practical settings, using very large block lengths may not be feasible due to complexity and delay constraints [8]. Furthermore, it's crucial to note that even under these assumptions, the separation theorem breaks down in multi-user scenarios [5], or for non-ergodic source or channel distributions [8]. These limitations highlight the importance of considering Joint Source-Channel Coding (JSCC) in practical communication systems.

Nevertheless, many upcoming situations, including applications like the Internet of Things, self-driving vehicles, and the tactile Internet, require the transfer of image and video data under strict constraints on time delay, data capacity, and energy consumption. These limitations make it impractical to use complex long-blocklength source and channel coding methods [1]. In practical scenarios, it has been observed that JSCC tends to have superior performance compared to when they are used separately [1][16]. JSCC algorithms consider both the source and the channel to design a coding scheme that is both small and can withstand errors. The research community has put forward several specific designs for JSCC. One approach in JSCC uses techniques like resource assignment, information interaction, and unequal error protection [14]. Another JSCC strategy combines source coding and channel coding into a single process to optimize the communication system [16]. Recently, due to its impressive performance in areas like computer vision, speech processing, and natural language processing, researchers have started employing Deep Learning (DL) methods to

support either source or channel coding [6].

DL is a new and powerful way to design JSCC schemes. Deep Neural Networks (DNN) can learn the complex relationships between the source, the channel, and the encoded message. This allows them to design JSCC schemes that are more efficient and robust than traditional methods. DL-based JSCC offers several advantages over traditional methods:

- **Adaptive Coding:** DL models can adapt to changing channel conditions and source statistics, making them suitable for dynamic communication environments [14].
- **End-to-End Design:** DL allows for end-to-end optimization of the communication system, considering both source encoding and channel encoding jointly [15].
- **Improved Performance:** DL-based JSCC schemes can achieve better compression efficiency and error resilience compared to traditional methods, especially in scenarios with strong source-channel interactions [10].

However, it is important to highlight that most previous research on JSCC has mainly focused on single-link scenarios. When a second hop is introduced, or in cases with multiple wireless hops (as seen in drone networks), the features of the channel undergo significant changes, bringing forth new possibilities and challenges. In this research, the emphasis will be on investigating different strategies suitable for two-hop channels. This includes examining decoding and re-encoding techniques, as well as exploring joint coding approaches, both with and without re-amplification.

The thesis is organized into six chapters, each with a specific purpose of presenting a thorough examination of the research topic. Chapter 1, the Introduction, provides a general overview of the study, highlighting its importance and goals. Chapter 2, Background, explores the fundamental concepts and relevant literature that support the research. In Chapter 3, State of the Art, a detailed examination of current developments and existing knowledge in the field is conducted. Chapter 4, Model, is dedicated to the development and description of the proposed model or methodology. Chapter 5, Results, presents the findings and outcomes derived from the application of the model. Finally, Chapter 6, Conclusion, summarizes the key insights, discusses their implications, and outlines avenues for future research.

2

Background

2.1 Simple Communication System

Claude Shannon's 1948 paper [11], "A Mathematical Theory of Communication," is a landmark in the field of information theory. It laid the foundation for the theoretical understanding of communication and has had a profound impact on the development of communication technologies. The concepts of source and channel encoding are now widely applied in various fields, including digital broadcasting, mobile communications, and computer networks.

Source encoding, also known as data compression, aims to reduce the redundancy inherent in the message to make it more compact and efficient for transmission. Shannon introduced the concept of entropy, which measures the average uncertainty or information content of a message. This concept allowed for the development of algorithms that can compress data without losing essential information. Channel encoding, also known as error-correction coding, aims to protect the encoded message from noise and interference that can occur during transmission.

Claude Shannon's groundbreaking research established fundamental principles in information theory. He demonstrated that the amount of information in a message can be quantified based on its surprise or unpredictability, introducing the concept of entropy. Furthermore, Shannon demonstrated that

reliable communication over noisy channels is possible without compromising data transmission speed. This insight led to the development of source and channel coding techniques, enabling nearly error-free communication as long as the data rate remains within the channel's capacity. Shannon's work laid the foundation for achieving optimal communication over noisy channels, as long as data transmission adheres to the channel's capacity constraints. His contributions, particularly in data compression (source coding) and error correction (channel coding), have become indispensable components of modern communication systems, influencing digital broadcasting, mobile communications, and computer networks.

The fundamental stages in the communication process include:

- The sender creates a message.
- The sender encodes the message to facilitate transmission.
- The encoded message is transmitted over a communication channel, such as a phone line or computer network.
- The message is decoded upon reaching the receiver.
- The receiver reproduces the original message.

Shannon's diagram remains relevant in contemporary times, serving as a foundation for the design and enhancement of communication systems.

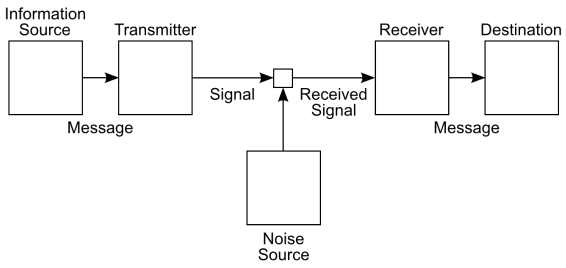

Figure 2.1: Block diagram of general communication system [11]

Modern communication systems are indeed intricate and multilayered structures, involving a diverse range of devices, entities, and software and hardware components operating over various physical media. They encompass a variety of information sources and endpoints, handling a vast spectrum of data types, transmission rates, and communication protocols. To manage this complexity, system designers adopt a modular approach where the design considerations for source coding are physically and logically separate from those for

channel coding. Each module in modern communication systems has a specific task, such as compressing the message, adding redundancy to the message, or correcting errors in the message. These modules are designed to be independent of one another, simplifying the design, construction, and maintenance of complex communication systems.

2.2 Communication Channels

Shannon's communication model highlights the significance of the channel, which is the pathway for transmitting information from the sender to the receiver. The process of source-channel coding transforms the source messages into signals that align with the channel's properties, underlining the channel's vital role in the design of the source-channel code. There are various channel models available, each unique in the physical conditions they represent, their abstraction level, and the simplifying assumptions they make.

Effective communication systems rely on the functionality and characteristics of communication channels, which facilitate the exchange of information through various mediums. In the realm of communication engineering and information theory, two fundamental channel models have emerged as significant components in this process: the Additive White Gaussian Noise (AWGN) channel and the slow fading channels.

The AWGN channel is a fundamental and widely used model in communication engineering and information theory. It is characterized by the presence of white Gaussian noise, which adds to the transmitted signal during transmission. This noise arises from random processes and exhibits a constant power spectral density, making it "white" across all frequencies. AWGN channels are often used to simulate real-world scenarios, where noise can corrupt the signal during transmission. The noise is represented as a vector of Independent Identically Distributed (I.I.D.) samples following a circularly symmetric complex Gaussian distribution. In mathematical terms, the transfer function of the AWGN channel is denoted as equation 2.1, where n represents the noise vector, and z is the input signal.

$$\eta_n(z) = z + n \tag{2.1}$$

In contrast, the slow fading channel accounts for the dynamic nature of wireless channels, influenced by factors like multipath propagation and environmental conditions. The widely adopted model for this scenario is the Rayleigh slow fading model. In this context, the impact of the channel on the transmitted signal is described as a multiplicative effect through the transfer function equation 2.2, where h is a complex random variable representing the channel gain.

$$\eta_h(z) = hz \tag{2.2}$$

To encapsulate the combined influence of channel fading and Gaussian noise, the transfer functions η_h and η_n are combined into a composite function, $\eta(z) = \eta_n(\eta_h(z))$, which reflects the overall distortion caused by the channel on the transmitted signal, encompassing the complexities of real-world communication channels.

$$\eta(z) = \eta_n(\eta_h(z)) = hz + n \tag{2.3}$$

2.3 Joint Source-Channel Coding (JSCC)

Shannon's separation theorem, a cornerstone of communication theory, asserts that source and channel coding can be optimized independently for memoryless and ergodic channels when using codes with infinite block lengths. However, this theoretical optimality comes with practical limitations [1].

The optimality of separation breaks down when dealing with non-ergodic source or channel distributions, which are common in real-world communication scenarios. Additionally, Achieving optimality using Shannon's approach requires infinitely long codewords, which are often impractical due to computational and memory constraints.

In light of these limitations, JSCC emerges as a promising alternative for practical communication systems. JSCC abandons the assumption of independent source and channel coding and instead considers them jointly, aiming to optimize the overall communication system rather than individual components.

This holistic approach has demonstrated significant gains in various schemes, including vector quantization and index assignment [16]. On the other hand, JSCC, which coordinates source and channel coding or combines them into a single step, may offer substantial improvements over the tandem coding approach.

2.4 Deep Learning

JSCC has long been recognized for its potential to outperform separate source compression followed by channel coding. However, the practical design of joint coding schemes has been a significant challenge due to the complexity involved in optimizing both source and channel coding simultaneously.

The emergence of DL has brought a paradigm shift in JSCC, enabling the development of more efficient and effective joint coding schemes. DL-based JSCC methods, such as deep JSCC, leverage the ability of DNN to extract complex features from training data while implicitly incorporating channel characteristics into their encoding process. This implicit incorporation of channel characteristics allows DL-based JSCC methods to adapt to various channel conditions without explicit channel modeling [9].

An autoencoder is a type of neural network that attempts to learn a compressed representation of its input data. It is similar to a traditional encoder-decoder model, but instead of trying to reconstruct the input exactly, the autoencoder aims to capture the essential features of the input. This compressed representation, known as the latent code, can then be used for various tasks, such as dimensionality reduction, anomaly detection, and image generation [3].

The autoencoder consists of two main components: an encoder and a decoder. The encoder takes the input data and transforms it into the latent code. The decoder then takes the latent code and attempts to reconstruct the original input. The autoencoder is trained by minimizing the difference between the original input and the reconstructed output.

Autoencoders are typically trained on unlabeled data, meaning that they do not require any prior knowledge of the data. This makes them a useful tool for unsupervised learning tasks, where the goal is to discover patterns and

structure in the data without any explicit guidance.

In the context of image compression using an autoencoder, the process involves feeding an input image into an encoder, which transforms the image into the latent code. This code retains the essential features of the image in a much smaller form. Subsequently, the latent code is fed into a decoder, which reconstructs the image. While the reconstructed image may not be an exact replica of the original, it encapsulates the most important features. This sequential transformation from input to compressed representation and then to reconstruction constitutes the fundamental mechanism of image compression using an autoencoder.

Autoencoders are a powerful tool for unsupervised learning that can be used for a variety of tasks. They are a relatively simple type of Neural Network (NN), but they can be surprisingly effective at learning complex representations of data. Autoencoders have a wide range of applications, including:

- Dimensionality reduction: Autoencoders can be used to reduce the dimensionality of data, which can be useful for data visualization and Machine Learning (ML) tasks.
- Anomaly detection: Autoencoders can be used to detect anomalies in data, such as fraudulent transactions or outliers in a dataset.
- Image generation: Autoencoders can be used to generate new images that are similar to the images they were trained on.

Convolutional Neural Networks (CNN) are a type of Artificial Neural Network (ANN) that are particularly well-suited for image recognition and other image-based tasks. CNNs were first introduced in the 1980s by Yann LeCun, but they have only recently become popular due to the availability of large datasets and powerful computing hardware [3].

CNNs work by using a series of filters to extract features from an image. These filters are convolved, or slid, across the image, and the results are combined to produce a feature map. The feature map is then passed to another layer of filters, and the process is repeated until a final classification is made.

CNNs have been shown to be very effective for a variety of image-based tasks, including image recognition, object detection, and image segmentation. They are now widely used in a variety of applications, such as facial recognition, medical imaging, and autonomous vehicles. One of the key advantages of

CNNs is that they are able to learn features from data without being explicitly programmed. This makes them well-suited for tasks where the features of interest are not well-defined, such as image recognition. CNNs are also very efficient in terms of computation. This is because they can take advantage of the locality of information in images. For example, a filter that is designed to detect edges will only need to be applied to a small region of the image.

Despite their many advantages, CNNs also have some limitations. One limitation is that they can be sensitive to noise in the input image. Another limitation is that they can be difficult to interpret. This is because the features that a CNN learns are not always easy to understand.

3

State of the Art

The paper [4] presents a study on the problem of joint source-channel coding for Multiple Input, Multiple Output (MIMO) block-fading channels. The authors considered a scenario where only the Channel State Information at the Receiver (CSIR) is available at the receiver. Therefore, Shannon's source-channel separation theorem does not apply, making a joint source-channel approach necessary. The authors focus on the high Signal to Noise Ratio (SNR) regime and define a figure of merit called the distortion exponent, which measures the rate at which the average distortion decreases with increasing SNR. They derive an upper bound on the distortion exponent and three different lower bounds. The authors also present three different transmission schemes, namely progressive superposition, hybrid digital/analog transmission, and progressive layered source coding. Each scheme is optimized for different bandwidth ratios and channel configurations. It is shown that progressive layered source coding with unequal error protection is a crucial technique for adapting to variable channel conditions without CSIR. For the proposed transmission schemes, the authors show that progressive or simultaneous transmission of the layers performs better, depending on the bandwidth ratio. For single-block Multiple Input, Single Output (MISO)/Single Input, Multiple Output (SIMO) channels, the authors show that the hybrid digital/analog transmission scheme outperforms all other strategies and meets the upper bound, implying that the distortion exponent is optimal for MISO/SIMO channels.

A new method for training autoencoders for lossy image compression is proposed in [12]. Autoencoders are a type of neural network that can be used to compress images by learning to encode them into a smaller representation and then reconstruct them from that representation. The proposed method is based on using a simple but effective way to deal with the non-differentiability of the compression loss. This non-differentiability makes it difficult to train autoencoders directly for lossy compression. The proposed method also uses an incremental training strategy, which helps to improve the performance of the autoencoder. The results show that the proposed method can achieve better performance than JPEG 2000 in terms of Structural Similarity Index (SSIM) and Mean Opinion Score (MOS) scores. This performance is achieved using an efficient convolutional architecture, combined with simple rounding-based quantization and a simple entropy coding scheme. The paper also discusses the advantages of using an end-to-end trained autoencoder for lossy image compression. An end-to-end trained autoencoder can be optimized for arbitrary metrics, which makes it possible to optimize the autoencoder for perceptual quality. The paper concludes by discussing future work, such as exploring the optimization of compressive autoencoders for different metrics and using Generative Adversarial Networks (GANs) to train autoencoders for lossy image compression.

In the paper titled "Deep Convolutional AutoEncoder-based Lossy Image Compression" a novel approach to lossy image compression is introduced. They utilize a Convolutional Autoencoder (CAE) to enhance coding efficiency, replacing traditional transforms and training the CAE with a rate-distortion loss function. To create a more energy-efficient representation, they employ Principal Component Analysis (PCA) to transform feature maps generated by the CAE, followed by quantization and entropy coding for compression. Their method outperforms traditional image coding algorithms, achieving a 13.7% BD-rate reduction compared to JPEG2000 for Kodak database images, while maintaining similar complexity. The authors plan to further improve their approach by incorporating perceptual quality metrics and exploring the use of GANs in future research [2].

In [1] a novel technique for wireless image transmission is introduced. This technique eliminates the need for explicit compression or error correction codes by combining source and channel coding. This method directly maps

image pixel values to complex-valued channel input symbols using two CNNs for encoding and decoding. The networks are jointly trained, resembling an autoencoder with an untrainable intermediate layer that simulates the noisy communication channel. Their results demonstrate the superiority of this deep JSCC scheme over traditional digital transmission methods that employ JPEG or JPEG2000 compression followed by channel coding, especially in low SNRs and limited channel bandwidth scenarios, and it exhibits a graceful performance degradation as channel SNR varies. Additionally, deep JSCC excels in the presence of slow Rayleigh fading channels, outperforming separation-based digital communication in various SNR and channel bandwidth conditions .

The authors highlight that the deep JSCC architecture is not limited to reliable channel communication but also efficient image compression. They emphasize the ability of the network to map salient features to nearby representations for noise-resilient reconstruction in the presence of channel noise. Furthermore, it acts as a regularizer for the autoencoder. In their future work, the authors plan to enhance the deep JSCC system by exploring more advanced NN architectures in the autoencoder to improve compression performance. They also aim to test the system's performance in non-Gaussian channels and channels with memory, where capacity-approaching channel codes are unavailable, expecting that the advantages of the proposed neural network-based JSCC scheme will be even more apparent in such non-ideal settings.

The authors of [7] address the challenge of image retrieval over wireless channels, especially in surveillance scenarios involving wireless cameras and drones. Traditional methods employ lossy image compression for query images to reduce data transmission over bandwidth and power-limited wireless links. However, the authors propose a novel approach by introducing a DNN-based compression scheme tailored for retrieval tasks, eliminating the need for full image reconstruction. Furthermore, they present a JSCC approach using DNNs, which not only enhances end-to-end accuracy but also simplifies and accelerates the encoding process, particularly advantageous for power and latency-constrained Internet of Things (IoT) applications. The authors conclude that their autoencoder-based JSCC scheme outperforms digital and JSCC schemes lacking feature decoding, highlighting the importance of DNN-based JSCC methods to address the stringent latency and bandwidth constraints in wireless image retrieval applications.

In the paper titled "DeepJSCC-f: Deep Joint Source-Channel Coding of Images With Feedback" [8] the authors tackle the challenge of wireless image transmission in the presence of channel output feedback. They recognize that, from a theoretical perspective, feedback doesn't enhance asymptotic end-to-end performance, and conventional methods using separate source coding followed by capacity-achieving channel coding are optimal. However, in practical finite block length scenarios, separation is not ideal. The authors introduce DeepJSCC-f, an autoencoder-based JSCC scheme that leverages channel output feedback. This innovative approach significantly improves end-to-end reconstruction quality for fixed-length transmission or reduces the average delay for variable-length transmission, surpassing traditional coding-based methods. It is a pioneering practical implementation of a JSCC scheme exploiting channel output feedback, showcasing how modern machine learning techniques can lead to efficient communication methods that outperform structured coding-based designs. Additionally, the authors highlight the scheme's adaptability, graceful degradation in varying channel conditions, and potential applicability to other data types beyond images, such as audio or video.

The limitations of deep learning-based JSCC for wireless communications are addressed in [14], which often operate within specific SNRs regimes, making them inefficient when dealing with varying SNR levels during deployment. To tackle this challenge, they introduce a novel approach called Attention DL-based JSCC (ADJSCC) inspired by traditional JSCC's resource assignment strategy. ADJSCC dynamically adjusts CR and channel coding rates according to channel SNR by employing attention mechanisms, which allocate computing resources effectively. Extensive experiments show that ADJSCC outperforms existing DL-based JSCC methods in terms of adaptability, robustness, and storage efficiency. It requires less storage and is more resilient in the presence of channel mismatch. In conclusion, the authors propose the ADJSCC method, which automatically adapts to various channel conditions and offers improved performance, computational complexity, and storage efficiency compared to existing approaches. They evaluate ADJSCC on various scenarios and datasets, demonstrating its adaptability and efficiency. Future work may include extending this method to high-definition images and real wireless channels, further promoting deep learning-based JSCC technology for practical wireless communication systems.

The paper [13] introduces a collection of neural network-based full-resolution lossy image compression techniques. These methods provide the flexibility to adjust compression rates during use without the need for retraining. The architectures include an Recurrent Neural Network (RNN)-based encoder and decoder, a binarizer, and a NN for entropy coding. The paper explores different RNN types (Long Short-Term Memory (LSTM), associative LSTM) and introduces a new hybrid of Gated Recurrent Unit (GRU) and ResNet. It also compares "one-shot" and additive reconstruction architectures, presenting a new scaled-additive framework. Their results demonstrate improvements of 4.3% to 8.8% Area Under Curve (AUC) in the rate-distortion curve, depending on the perceptual metric used when compared to previous work. Notably, their models outperform JPEG for image compression across most bitrates on the Kodak dataset, both with and without entropy coding. In the discussion, the authors acknowledge the challenge of selecting a single best model due to the varying performance of their models based on different perceptual metrics. They emphasize the benefit of adding entropy coding, particularly in the early iterations where recurrent encoder models generate spatially correlated codes. The authors plan to continue their work by competing with compression methods derived from video codecs and by exploring joint training of the entropy coder and patch-based encoder. They also highlight the evolving nature of perceptual differences and the need for better metrics that correlate with human vision for all types of distortions.

4

Model

4.1 Problem Defenition

Wireless communication networks are now an essential part of modern society, making it easy to send and receive multimedia data like images, videos, and audio. Even though these networks are everywhere, they face challenges with the quality of the data they send. This is often caused by noise and interference in the wireless channels. To solve this problem, researchers have turned to JSCC, which is a promising technique that aims to keep the quality of multimedia data high through semantic communication.

Although most of the research on JSCC has focused on single-hop communication links, where data is sent directly from the source to the destination, wireless networks often use multi-hop communication. One particular scenario is the two-hop wireless communication network, which involves two communication links between the source and the destination. This configuration is common in a variety of applications, such as multi-hop wireless ad hoc networks, sensor networks, and emerging technologies such as drone networks. The addition of an intermediate node in this setup makes it more difficult to maintain data quality and manage the effects of channel noise. In these cases, data travels through multiple nodes before reaching its final destination, which creates a more complex channel scenario and introduces new opportunities and

challenges for JSCC. In this research we explore extending JSCC to two-hop wireless communication networks by investigating solutions such as decoding and re-encoding, joint coding with re-amplification, and joint coding with mapping. The study aims to provide valuable insights into the adaptability and effectiveness of these techniques in more complex communication scenarios.

4.2 Decoding and Re-encoding (DREC)

The DREC model is a complex system that works in a two-hop wireless communication network. It uses two autoencoders in a cascade to achieve high-quality data transmission. The model has several parts, including two encoders, two decoders, and two communication channels. These parts work together to send and rebuild encoded data.

The DREC model shown in Figure 4.1 works as follows:

1. The input data is first encoded by the first encoder.
2. The encoded data is then transmitted through the first communication channel.
3. Upon reaching the end of the first communication channel, the encoded data is decoded by the first decoder.
4. The decoded data is then passed to the second encoder for a secondary encoding.
5. The second encoder encodes the decoded data and transmits it through the second communication channel.
6. The second decoder decodes the received data and reconstructs the original image.

The DREC is flexible because its two autoencoders can be either identical or different. This allows the model to be adapted to different needs and requirements. For example, if the two communication channels have different bandwidth or noise characteristics, different autoencoders can be used to improve the model's performance.

When the autoencoders are similar, each node can decide whether to be a sender, receiver, or intermediate node. This flexibility makes the model easier to use, as a single model design and training process is sufficient for all nodes. This reduces additional costs such as memory and computational capacity.

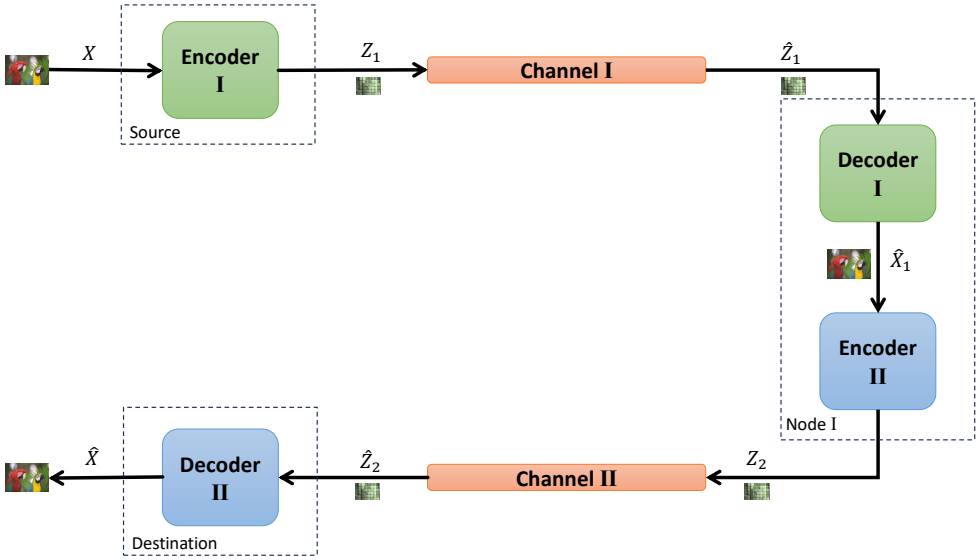

Figure 4.1: Decoding and Re-encoding (DREC) model

One practical enhancement that can be incorporated into the DREC model is to add an enhancement block between first decoder and second encoder. This additional block would improve the image quality before the decoded image is fed into the second encoder. While this would incur some additional costs, it would also present an opportunity to improve the overall performance of the communication process.

The DREC model has many advantages, but it also has some potential challenges. One of the main challenges is security and privacy. Nodes between the two communication channels can decode and encode images, which could lead to unauthorized access to image content. Another security and privacy concern is the potential for data leakage. The DREC model involves double encoding and decoding images, which could potentially introduce new vulnerabilities that could be exploited by malicious actors to steal or modify the transmitted data. Additionally, the double encoding and decoding processes can increase latency.

This research employs autoencoders that are grounded in the framework put forth by Bourtsoulatze et al. (2019) [1]. The JSCC scheme they suggested is depicted in Figure 4.2. Using a JSCC approach, the transmitter transforms the input image x, represented by n real numbers, into a vector of k complex-valued

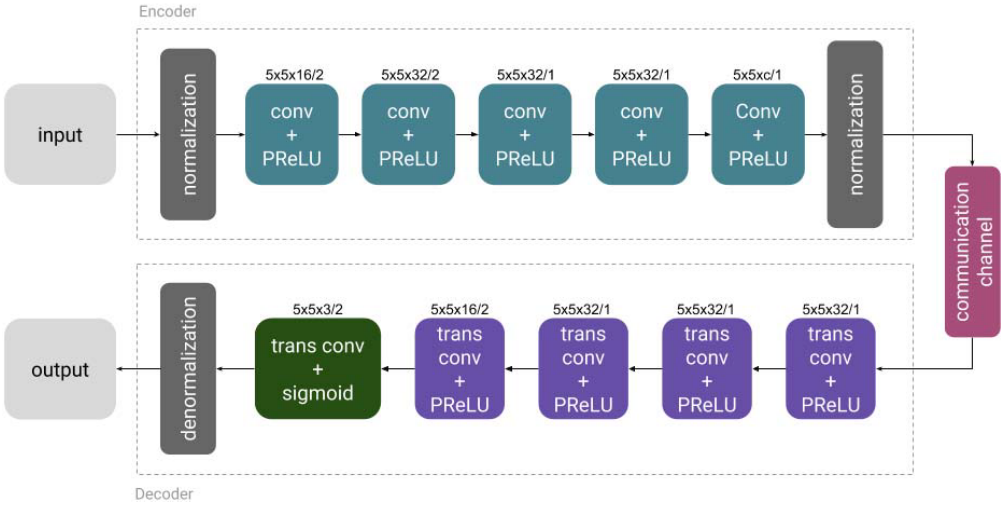

Figure 4.2: Architecture of the encoder and decoder for the JSCC scheme proposed in [1].

channel input symbols z. In line with JSCC terminology, the image dimension n is referred to as the source bandwidth, while the channel dimension k is termed the channel bandwidth. Since k is typically smaller than n, this process is known as bandwidth compression. The ratio k/n is defined as the bandwidth compression ratio. In real-world communication systems, practical limitations such as energy constraints and interference necessitate that the transmitter's output adheres to specific power constraints, including peak and average power limits. The average power constraint is formulated as follows:

$$\frac{1}{k}E[z^* \cdot z] \leq P \tag{4.1}$$

This change is made possible by a specific encoding function, represented as $f_\theta : \mathbb{R}^n \rightarrow \mathbb{C}^k$, which is directed by a group of parameters known as θ. The function of the encoder, f_θ, is carried out using CNN, which also uses the parameters θ.

The structure of the CNN in the encoder is made up of a series of convolutional layers. Each of these layers is followed by a Parametric Rectified Linear Unit (PReLU) activation function and a normalization layer. Every part of this system has a specific job. The main job of the convolutional layers is to pull out important features from the input image. These features are key for the coding process that comes next. These features are then put together to make the

channel input samples. The PReLU activation functions allow for a nonlinear mapping from the source signals domain to the coded signals domain. The final result, $\tilde{z} \in \mathbb{C}^k$, which comes from the last convolutional layer of the encoder, goes through a normalization process.This normalization process is described as follows:

$$z = \sqrt{k \cdot P} \times \left(\frac{\tilde{z}}{\sqrt{\tilde{z}^* \cdot \tilde{z}}} \right) \tag{4.2}$$

In this scenario, $\tilde{z}^*$ is the conjugate transpose of $\tilde{z}$. This makes sure that the channel input, z, follows the set average transmit power limit, P. By using this normalization process, we can keep the average power limit in check within the transmitted channel input samples. This way, we meet the required power standards.

Once the encoding process is complete, the combined source-channel coded sequence, denoted as z, is sent over the communication channel. This is done by directly using the real and imaginary parts of the channel input samples across the In-phase (I) and Quadrature (Q) components of the digital signal. However, the channel introduces random distortions to the transmitted symbols, which is represented by $\eta : \mathbb{C}^k \rightarrow \mathbb{C}^k$.

The receiver includes a joint source-channel decoder. This decoder uses a decoding function, $g_\phi : \mathbb{C}^k \rightarrow \mathbb{R}^n$, to map the distorted complex-valued signal, $\hat{z} = \eta(z) \in \mathbb{C}^k$, to an estimate of the original input, $\hat{x} \in \mathbb{R}^n$. The decoding function, like the encoding function, is parameterized by the decoder CNN with the parameter set ϕ. The NN decoder undoes the operations carried out by the encoder. It does this by passing the received (and possibly distorted) coded signal, $\hat{z}$, through a series of transpose convolutional layers (which include non-linear activation functions). This maps the image features to an estimate, $\hat{x}$, of the image that was originally transmitted.

For a comprehensive end-to-end optimization of the communication system, it's imperative to integrate the communication channel into the overall NN architecture. The communication channel is represented as a series of non-trainable layers, depicted by the transfer function $\hat{z} = \eta(z)$. A slow fading channel with AWGN is modeled using equation 2.2. Although it's feasible to

employ distinct layers for each of the autoencoders and their internal blocks, this research opts for a consistent structure across all components.

The second autoencoder has the flexibility to be either similar to the first one or have a completely different structure. The same goes for the second channel - it can be identical or different from the first one. However, in this study, both the autoencoders and channels are considered to be similar.

In the DREC model, the core objective revolves around minimizing the average distortion between the original input image (denoted as x) and its reconstruction ($\hat{x}$) produced by the decoder. This process involves optimizing the encoding and decoding functions, where different approaches for this minimization can be considered.

Joint Minimization Approach: One way to handle this minimization is to view the entire network, comprising encoders, channels, and decoders, as one interconnected system. This holistic perspective leads to the minimization of the joint probability distribution ($p(x, \hat{x})$) of the original and reconstructed images, aiming to minimize the distortion measure ($d(x, \hat{x})$). Mathematically, this can be expressed as:

$$(\theta^*, \phi^*) = \arg\min_{\theta,\phi} \mathbb{E}_{p(x,\hat{x})}[d(x, \hat{x})] \tag{4.3}$$

However, this method faces a challenge often, the true distribution of the input data ($p(x)$) remains unknown. Consequently, an analytical form of the expected distortion in 4.3 is also indeterminable due to the unknown probability distribution.

Independent Autoencoder Approach: Alternatively, one can treat each autoencoder as a separate and independent network, leading to individual minimization functions for each. This results in distinct minimization objectives:

$$(\theta_1^*, \phi_1^*) = \arg\min_{\theta_1,\phi_1} \mathbb{E}_{p(x,\hat{x}_1)}[d(x, \hat{x}_1)] \tag{4.4}$$

$$(\theta_2^*, \phi_2^*) = \arg\min_{\theta_2,\phi_2} \mathbb{E}_{p(x,\hat{x})}[d(x, \hat{x})] \tag{4.5}$$

In 4.4, the focus is on the first autoencoder, optimizing the distortion measure between the original image x and the output of the first decoder ($\hat{x}_1$). In 4.5, the second autoencoders objective is to minimize the difference between the final decoded image ($\hat{x}$) and the original image (x), even though its input is $\hat{x}_1$, the output of the first decoder.

Its important to note that the distortion measure in 4.5 for the second autoencoder is strategically designed to facilitate tuning. The measure serves as a guide, ensuring that the second encoder and decoder are adjusted in a manner that minimizes the distortion compared to the original image. This approach assumes that by measuring the discrepancy between the final decoded image and the original input, the second autoencoder can adapt to generate output that closely resembles the original image.

Choosing the Optimal Approach: The choice between joint optimization and independent optimization depends on several factors, including the complexity of the network, the availability of computational resources, and the desired level of performance. In the joint approach, the entire network needs training for different combinations of two channel types. On the other hand, the independent approach offers flexibility. It involves developing separate autoencoders for different channel models and combining two based on the preferred channel condition. In this thesis, the independent autoencoder approach is employed.

4.3 Joint Coding With Re-amplification (JCWR)

JCWR is a model for two-hop wireless communication networks that uses a single autoencoder. In this model, the encoder acts as the sender, and the decoder acts as the receiver. The unique feature of JCWR is that it uses an amplifier at the end of the first communication channel (Figure 4.3). This amplifier boosts the noisy signal before it is transmitted through the second communication channel. The decoder at the end of the second communication channel then processes the amplified signal and decodes it to reconstruct the original image.

It is possible to implement JCWR in a way that enables each node to act as a sender, receiver, or amplifier. This flexibility depends on the presence of all

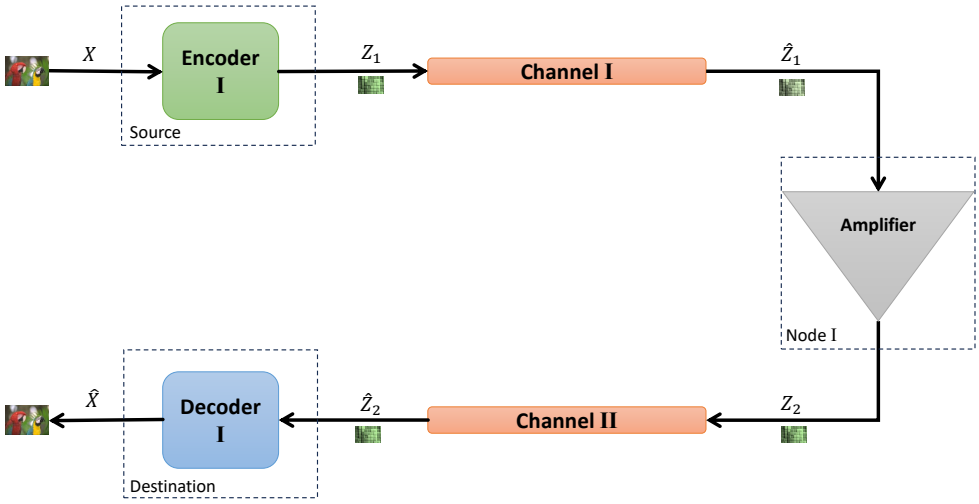

Figure 4.3: Joint Coding With Re-amplification (JCWR)

three functional blocks in each node. Conversely, assigning a specific role to each node introduces an alternative dimension of flexibility. In this scenario, nodes can have a more profound network structure for their designated roles with the same memory and computational capacity. As a result, this configuration has the potential to improve the overall quality of data at the final destination.

In situations where each node can do all three jobs, the system is very adaptable. It can easily switch roles. For example, a node could start as a sender in one session and then act as an amplifier or receiver in the next session. This adaptability makes sure resources are used well and can make the network stronger when communication needs change.

On the other hand, giving each node a specific role makes the network more structured. Each node becomes an expert in one job, making the network run smoother. This can be good in situations where certain nodes need to always do the same job. For instance, in a surveillance network, some nodes may always act as senders, while others are always receivers or amplifiers.

It is important to note that the amplifier in the JCWR model only processes encoded data and does not have access to the actual content of the image. This feature helps address security and privacy concerns that may arise during the transmission process. The inherent separation of the amplifier from the

image content ensures a level of data privacy, as the amplifier only interacts with encoded information rather than the explicit image content. This is especially important in applications where the data being sent may be private, like in healthcare, surveillance, or military communications.

The JCWR model is designed to be simple, with a single autoencoder responsible for both encoding and decoding tasks. The amplifier can be either a straightforward device that re-sends its input with increased power or a more intricate mechanism. The latter option enables the utilization of advanced transmission and signal enhancement techniques, tailored based on the unique characteristics of the communication channels. This compensation mechanism ultimately aims to elevate the overall quality of the final image at its destination.

The JCWR model may have limitations, especially when the noise in the first communication channel is severe. In these cases, amplifying the signal alone may not be enough to overcome the complexities of complex interference patterns. The model's effectiveness depends on the amplifier's ability to sufficiently neutralize noise and boost the signal, which can be difficult in environments with significant channel-induced distortions.

In this model, the encoder and decoder are constructed following the structure outlined in the DREC model. The amplifier is incorporated as an untrainable layer throughout the entire network. The entire structure is trained as a single network with the objective of minimizing the distortion between the input image x and the final output $\hat{x}$. The optimization goal, as specified in equation 4.3, is to ensure that the original image x closely aligns with the final received image $\hat{x}$. Despite the potential for more sophisticated amplifiers with noise-filtering capabilities, this study focuses on the effectiveness of a basic amplifier that amplifies all inputs by a constant gain factor.

4.4 Multi-Link models

In this section, the scope is expanded to multi-link scenarios where there are more than two communication channels between the sender and receiver. In practical situations, having additional channels introduces complexity, and each channel may exhibit different characteristics and SNRs. As the number of links increases, the combinations of parameters among all nodes grow exponentially,

making it impractical to investigate every situation. Despite these challenges, it is valuable to explore certain combinations. To address this, the DREC and JCWR models are extended to accommodate three channels and two nodes between the source and destination. For simplicity and comparability with the previous two-link models, all nodes and channels are considered identical. The expanded models are illustrated in Figure 4.4 and Figure 4.5.

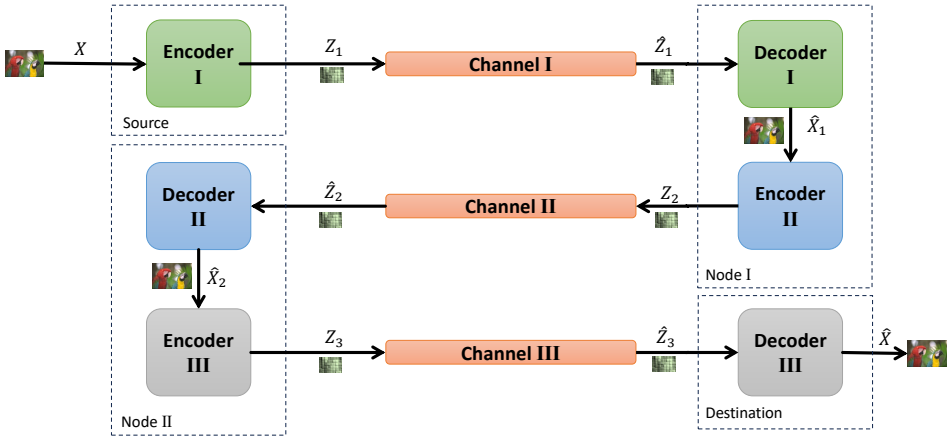

Figure 4.4: DREC model with three channel links between source and destination.

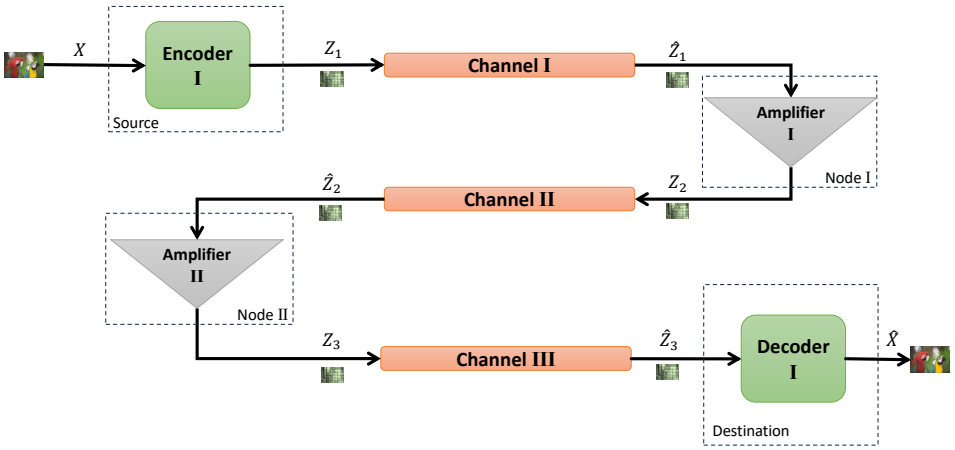

Figure 4.5: JCWR model with three channel links between source and destination.

5 Results

5.1 Evaluation Method

In this study, the effectiveness of the proposed models is evaluated using the COCO 2017 image dataset. Although the COCO dataset is extensive and covers object detection, segmentation, key-point detection, and captioning, this research doesn't utilize its annotations. The dataset consists of 118,000 images for training, 5,000 for validation, and 41,000 for testing. For this study, a subset of 50,000 training images and 10,000 test images are randomly chosen. To standardize the data, all images, which vary in size, are resized to 64x64 pixels.

Since the decoder typically lacks knowledge about the input data's statistics, the input images are normalized by the maximum pixel value of 255. This normalization process ensures that pixel values fall within the range of [0, 1].

The final output from the last convolutional layer in the encoder comprises $2k$ real values. To transform these values into k complex values, the first k values are interpreted as the real part, while the subsequent k values are perceived as the imaginary part. Conversely, the output from the channels is converted from the complex space back to the real space using the inverse process.

All the models were developed using PyTorch. During the training

phase, the Adam optimization algorithm was employed, with a learning rate set to 10^{-3}. The loss function used was the average Mean Square Error (MSE) between the original input image (x) and the reconstructed image ($\hat{x}$) at the decoder's output. This loss function is denoted as L and is defined by equation 5.1. The MSE distortion, $d(x, \hat{x})$, which is a measure of the difference between the original image (x) and the reconstructed image ($\hat{x}$), is defined in equation 5.2. In simpler terms, it calculates the average squared difference between the corresponding pixels of the two images. This formula aligns with the loss function described for each model in chapter 4.

$$L = \frac{1}{N} \sum_{i=1}^{N} d(x_i, \hat{x}_i) \tag{5.1}$$

$$d(x, \hat{x}) = \frac{1}{n} ||x - \hat{x}||^2 \tag{5.2}$$

The performance of all models and benchmark schemes is assessed using the Peak Signal-to-Noise Ratio (PSNR) metric. PSNR measures the quality of a reconstructed image in comparison to the original image. A higher PSNR value signifies a better quality reconstruction. The PSNR metric is calculated as:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \text{ (dB)} \tag{5.3}$$

Where:

- MSE is the average squared difference between corresponding pixels in the original and reconstructed images.
- MAX represents the maximum possible value of image pixels. For 24-bit RGB images (8 bits per pixel per color channel), MAX is 255.

Consequently, a higher PSNR value suggests a lower MSE, indicating a closer resemblance between the reconstructed and original images.

The Channel SNR (CSNR) measures the proportion between the average power of the intended signal (in this case, the coded image) and the average

power of the unwanted noise. The calculation is expressed as:

$$\text{SNR} = 10 \cdot \log_{10}\left(\frac{P}{\sigma^2}\right) \text{ (dB)} \tag{5.4}$$

Where:

- SNR represents the CSNR in decibels (dB).
- P denotes the average power of the coded signal (channel input signal).
- σ^2 signifies the average noise power.

A higher SNR value suggests a stronger signal and lower noise level, indicating a greater likelihood for the signal to be accurately detected and decoded.

All the schemes' performance is evaluated under the premise of a slow Rayleigh fading channel with AWGN. In this scenario, the channel transfer function is expressed as $\eta(z) = hz+n$, where $h \sim \mathcal{CN}(0, H_c)$ and $n \sim \mathcal{CN}(0, \sigma^2 I_k)$.

In this experiment, we do not make any assumptions about channel state information at either the receiver or the transmitter, and the transmission of pilot signals is not taken into account. Assuming slow fading, the channel gain h is randomly selected from the complex Gaussian distribution $\mathcal{CN}(0, H_c)$ for each transmitted image. This value remains constant throughout the entire transmission of the image and changes independently for each subsequent image. H_c is set to 1, and the noise variance σ^2 is adjusted to simulate varying average CSNR. During training, both channels' SNR is assumed to be identical. The channels are also considered identical during testing, except when explicitly mentioned.

All models were trained for a variety of CRs and SNRs. The CRs used were $\frac{1}{12}$, $\frac{1}{6}$, $\frac{1}{3}$, and $\frac{1}{2}$, while the SNRs used were 0 dB, 10 dB, and 20 dB. The input images were 64×64 pixels in size, and the value of k, which is related to the desired CR, was set to 4, 8, 16, and 24 for the respective CRs. During training, the SNR of both channels was considered to be the same. The training loop was run for 100 epochs.

After the training process is complete, the trained model's performance is assessed using a test dataset. To mitigate the effect of randomness introduced by the communication channel, each testing image from the test dataset is passed

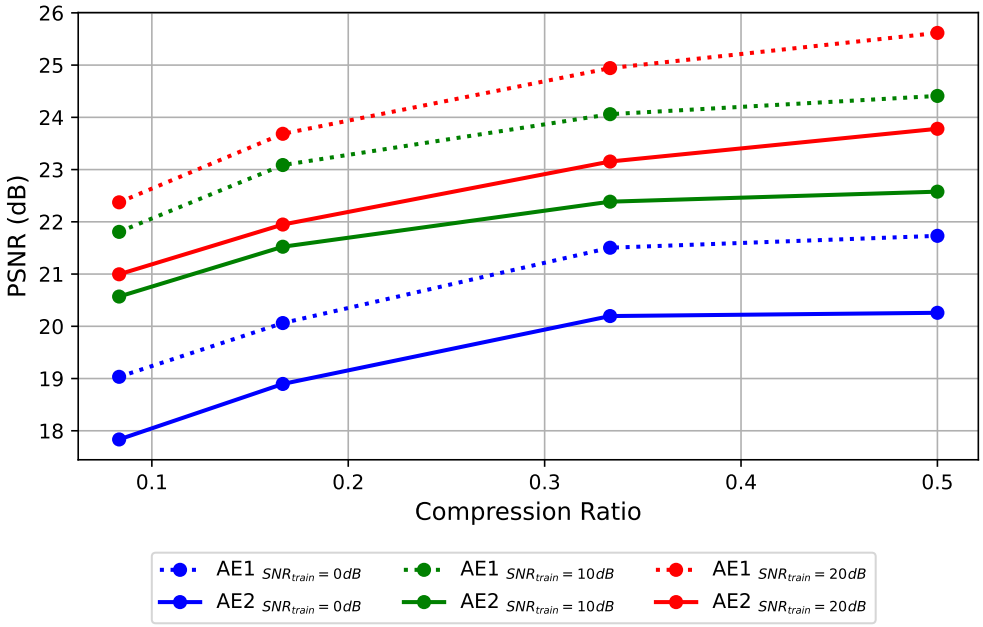

Figure 5.1: Performance of the trained DREC model with respect to compression ratio for different CSNR_{train} values (CSNR_{test}=CSNR_{train})

through the trained networks ten times, and the average PSNR of the received images is calculated. This process helps to smooth out the effects of noise and interference, providing a more accurate measure of the network's performance.

5.2 Decoding and Re-encoding (DREC)

Figure 5.1 displays the performance of the trained DREC model on test images concerning CRs for various CSNR_{train} values. All curves associated with AE2 pertain to the received images at the destination. Additionally, the figure illustrates the performance of AE1 to showcase the impact of the second channel. The second channel leads to approximately a 1 dB decrease in PSNR for lower CRs (higher compression) and around a 2 dB decrease for higher CRs. Moreover, it is evident from Figure 5.1 that training with lower SNR exhibits more robustness against the second channel. Conversely, the decrease in PSNR for SNR_{train} = 0 dB remains nearly constant across CRs, while for SNR_{train} = 10 dB and SNR_{train} = 20 dB, it increases with decreasing compression levels.

In Figure 5.2, the performance of DREC on test images concerning CRs for various CSNR_{test} values is depicted. For CSNR_{train} values of 10/20 dB,

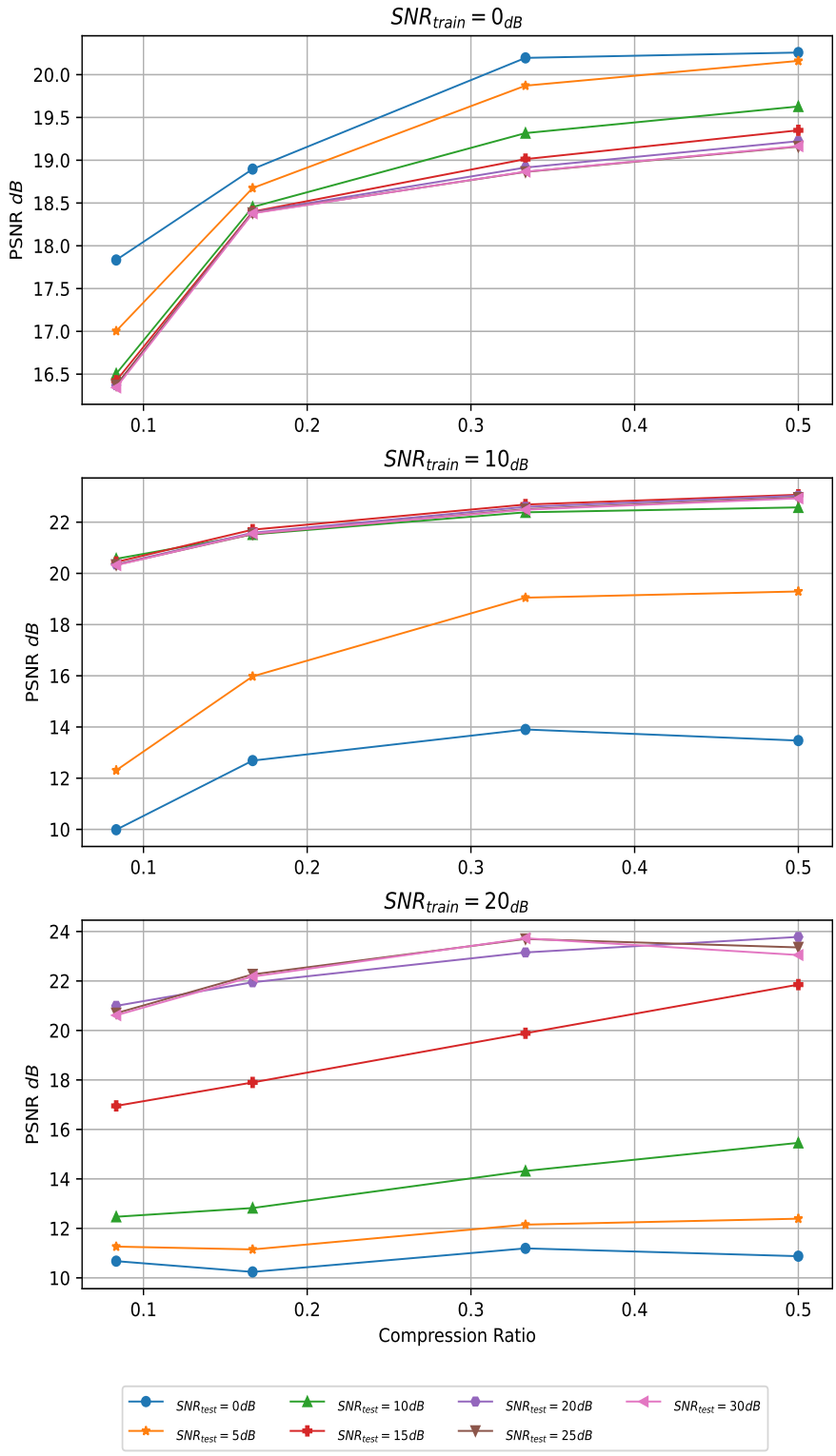

Figure 5.2: Performance of the trained DREC model with respect to compression ratio for different CSNR$_{test}$ values

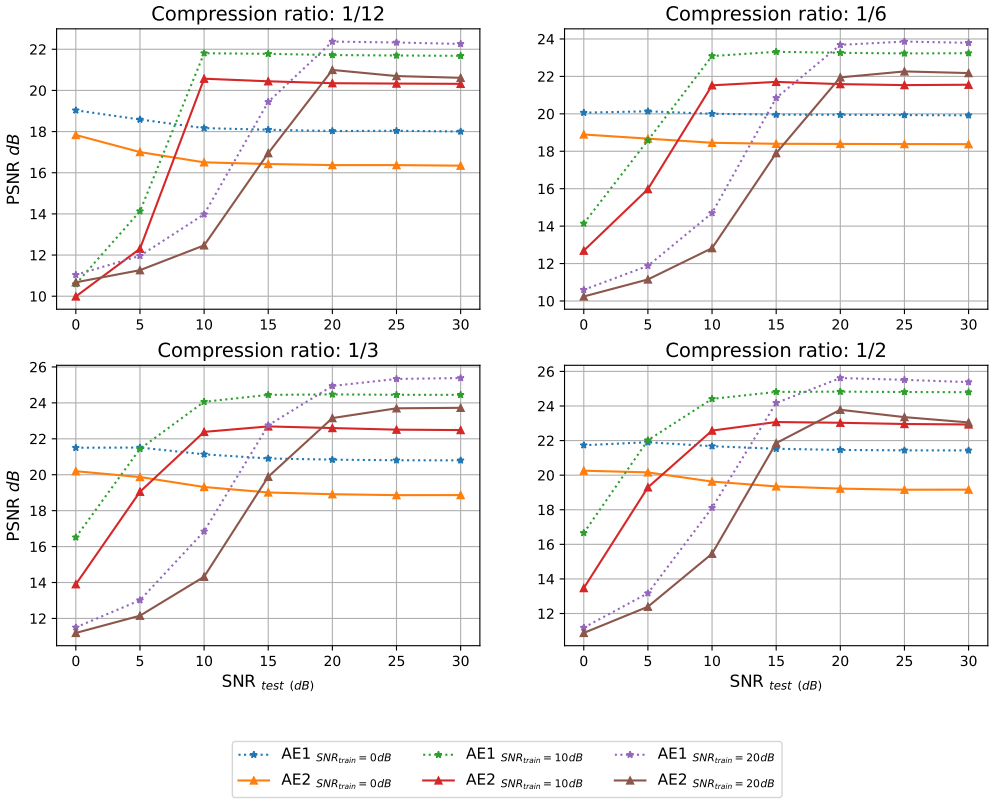

Figure 5.3: Performance of the trained DREC model with respect to different CSNR_{test} values for different CRs

there is a 50% decline in performance when CSNR_{test} is less than CSNR_{train}, reducing from 20 dB to 10 dB. Conversely, for CSNR_{train} of 0 dB, this decline is 1.5 dB. However, when CSNR_{test} is greater than or equal to CSNR_{train}, the overall performance remains nearly the same as when CSNR_{test} equals CSNR_{train}. This trend is further evident in Figure 5.3 and Figure 5.4, which illustrate the performance of DREC on test images considering CSNR_{test} values for various CRs and the performance of DREC on test images concerning CSNR_{test} values for different CSNR_{train} values.

The results support the model design and training approach. As explained earlier, AE1, the first autoencoder, is trained on the original images from the training set, while AE2, the second autoencoder, is trained on the output of AE1. Because AE1's decoding is not perfect and error-free, AE2 is trained on noisy reconstructed images. As a result, the errors caused by the first channel and the imperfect encoding-decoding steps tend to carry over to the second AE, leading to an expected drop in quality. However, the observations show that the

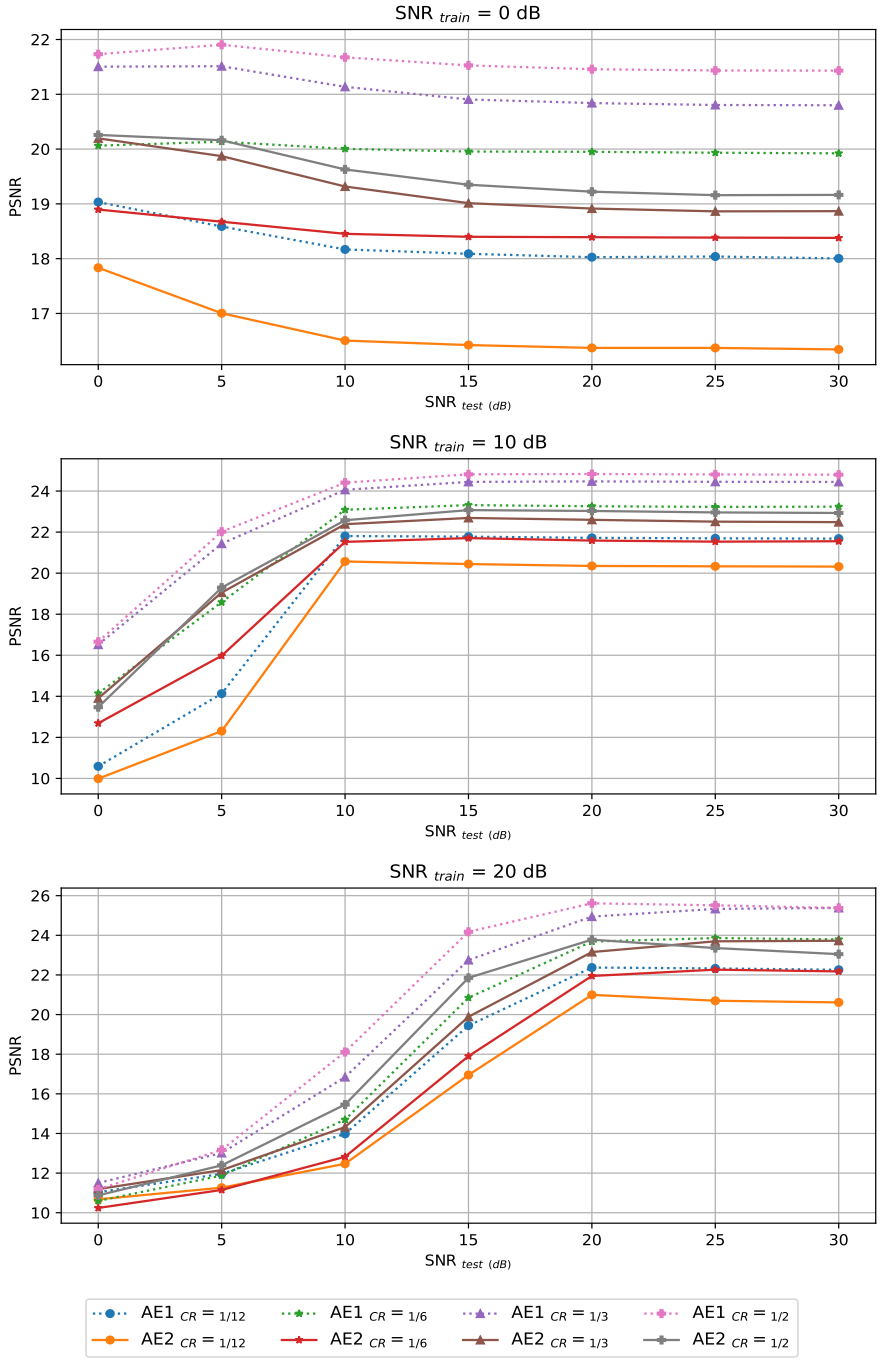

Figure 5.4: Performance of the trained DREC model with respect to different CSNR_{test} values for different CSNR_{train}

degradation is not too significant.

In addition, the results show that the model can handle changes in channel quality. Specifically, when SNR_{test} is lower than SNR_{train}, which means the channel conditions are worse than those for which the encoder/decoder were originally designed, the model is resistant to changes in channel quality. It shows a gradual decline in performance as the channel gets worse. In other words, the model is robust to the "cliff effect" seen in digital systems. In digital communication, the "cliff effect" occurs when the performance of a system suddenly drops sharply when the received signal quality becomes too weak. It is as if the system works well up to a certain strength or quality of the signal, but after that point, there is a sudden and severe decrease in performance [1] [8]. This effect is especially important in systems that use digital modulation and encoding, like wireless communication networks.

5.3 Joint Coding With Re-amplification (JCWR)

In Figure 5.5, we illustrate the performance of JCWR on test images concerning CRs for various amplification factors and CSNR_{train} values. The figure indicates that amplifying the received signal between the two channels is more beneficial when CSNR_{train} is lower. For instance, with CSNR_{train} at 0 dB, amplifying the signal by a factor of 2 increases the final PSNR by 2 dB, and by a factor of 4, it increases by 3 dB (for CR=$\frac{1}{12}$). However, this results in a decrease of 0.3 dB for an amplification factor of 2 and an increase of 0.8 dB for an amplification factor of 4. Nevertheless, for higher CRs, the impact of amplification is less pronounced.

To investigate the impact of amplification based on CSNR_{test}, Figure 5.7 is presented. Each row corresponds to one CSNR_{train} value, and each column represents one amplification factor. Amplification generally enhances the final performance (the same color on each row from left to right). The improvement is more noticeable when CSNR_{test} is lower than CSNR_{train} (different color on each plot). For instance, with CSNR_{train} at 20 dB (bottom row), for CR=$\frac{1}{12}$ and CSNR_{test} at 0 dB (blue curve), amplification factor 4 improves the final PSNR from 10 dB (left plot) to 13 dB (right plot). However, this improvement for CSNR_{test} at 30 dB (pink curve) is less than 1 dB. This enhancement is more

noticeable when CSNR_{test} is less than CSNR_{train}, and is more evident in this figure and Figures 5.6 and 5.8.

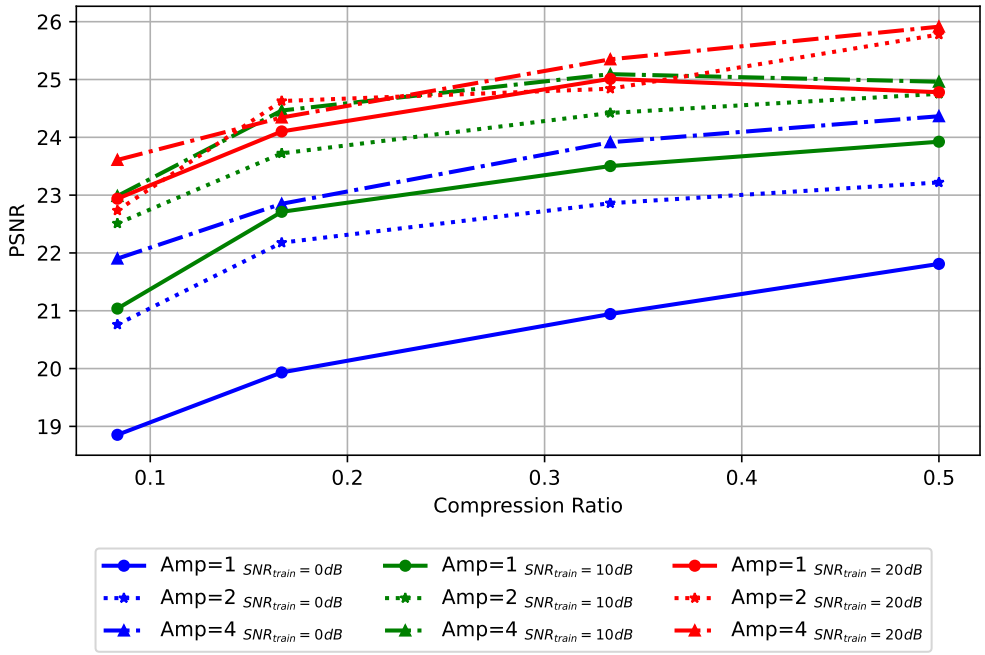

Figure 5.5: Performance of the trained JCWR model with respect to different amplification factors for different CSNR_{train} values (CSNR_{test}=CSNR_{train})

Figure 5.9 illustrates the shift in the performance trend of the entire system when changing the amplification factor from 1 (no amplification) to 4. Amplification makes both the encoded signal and noise stronger and the whole network tries to learn how to balance them out. To gain a deeper understanding of the signal propagation through two channels with an intervening amplifier, we can represent the transfer function of the second channel with an amplifier as $\hat{z}_2 = \eta_T(z_1)$, as expressed in Equation 5.7. Here α denotes the amplification factor (refer to Figure 4.3).

$$\hat{z}_1 = h_1 z_1 + n_1 \tag{5.5}$$

$$z_2 = \alpha \hat{z}_1 = \alpha h_1 z_1 + \alpha n_1 \tag{5.6}$$

$$\hat{z}_2 = h_2 z_2 + n_2 = (\alpha h_2 h_1) z_1 + (\alpha h_2 n_1 + n_2) = h_T z_1 + n_T \tag{5.7}$$

These equations suggest that the combination of channel 1, the amplifier, and channel 2 behaves similarly to a single channel but with new noise properties. During training, the entire network adapts to these characteristics, explaining the observed shift-like behavior. However, it's important to note that networks can train until these new noise characteristics make it impossible to understand the signal.

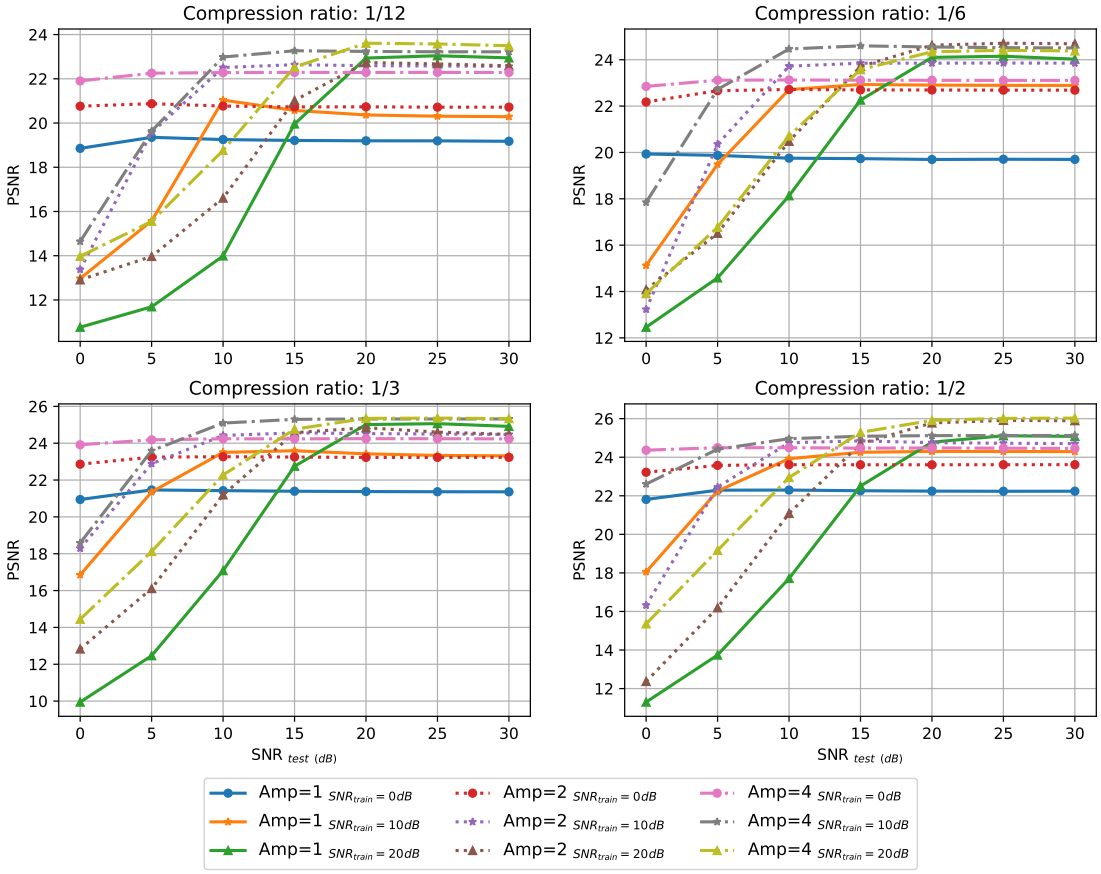

Figure 5.6: Performance of the trained JCWR versus CSNR_{test} for different CR values

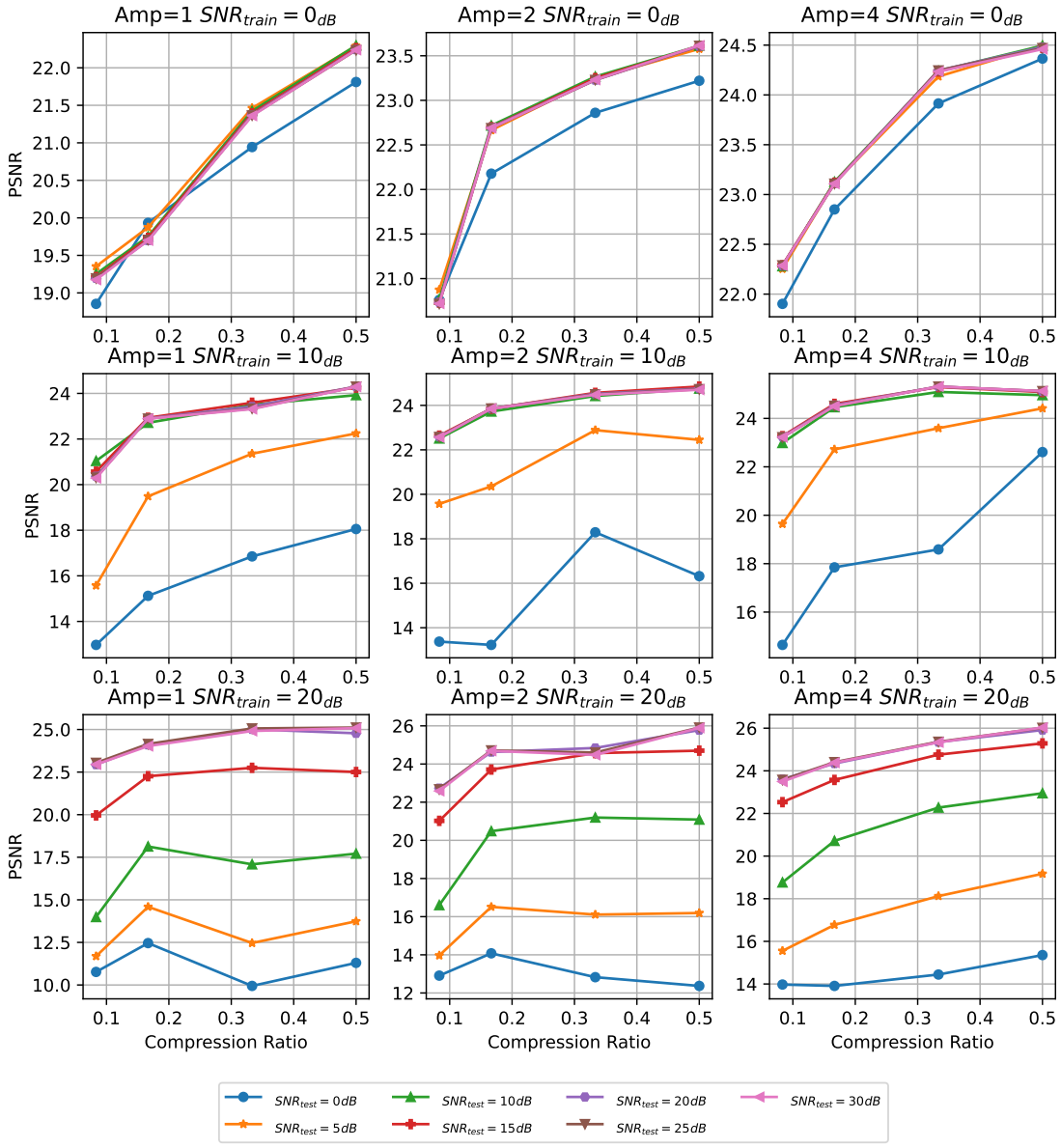

Figure 5.7: Performance of the trained JCWR model with respect to different amplification factors for different CSNR_{train}

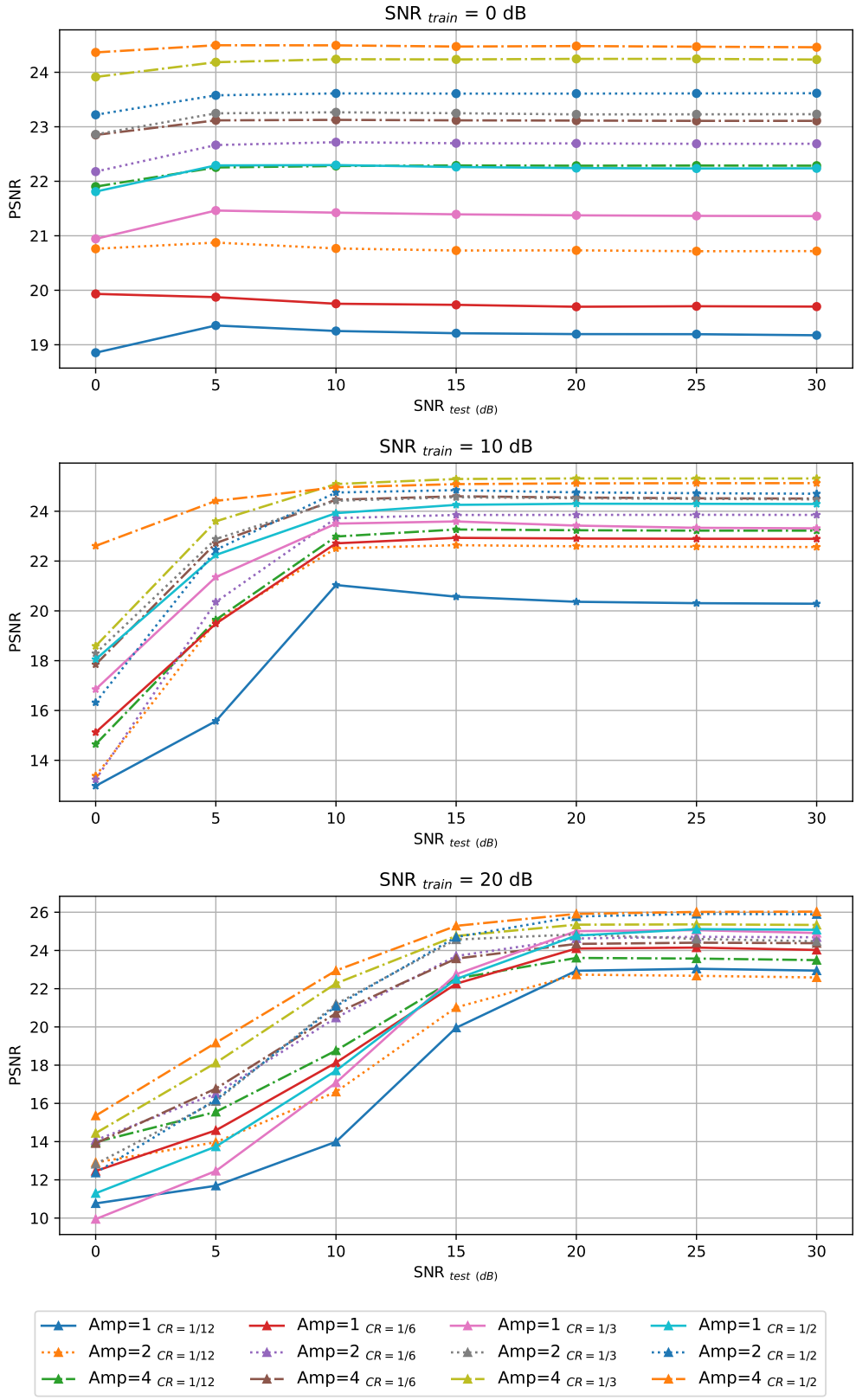

Figure 5.8: Performance of the trained JCWR for each CSNR_{train} for combinations of CR and CSNR_{test}

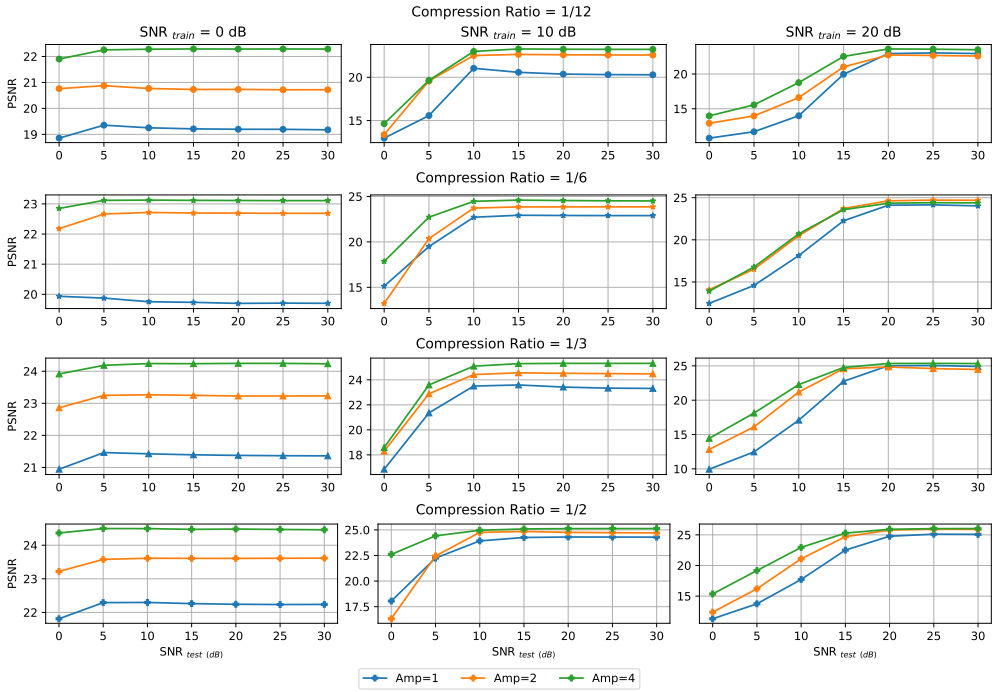

Figure 5.9: Performance of the trained JCWR for each CR and CSNR$_{train}$ concerning CSNR$_{test}$ and amplification factor

5.4 Multi-Link Models

To compare the impact of having more channels between two end nodes of the network for DREC and JCWR models, the same training conditions as the two-channel modes are followed. For both models, the previously trained two-link models are used as the starting point, and they are further trained for an additional 50 epochs.

For the 3-channels DREC model, an additional autoencoder is created. All three autoencoders of the model are initialized with the trained parameters of AE1 from the previously trained DREC model. Then, they are trained independently for 50 additional epochs. As depicted in Figure 4.4, the input of each encoder is the decoded image from the previous decoder. The first encoder is fed with the original image, and the loss function of each autoencoder calculates the difference between its output and the original image, similar to the DREC model.

For the 3-channels JCWR model, three channels with two amplifiers, one

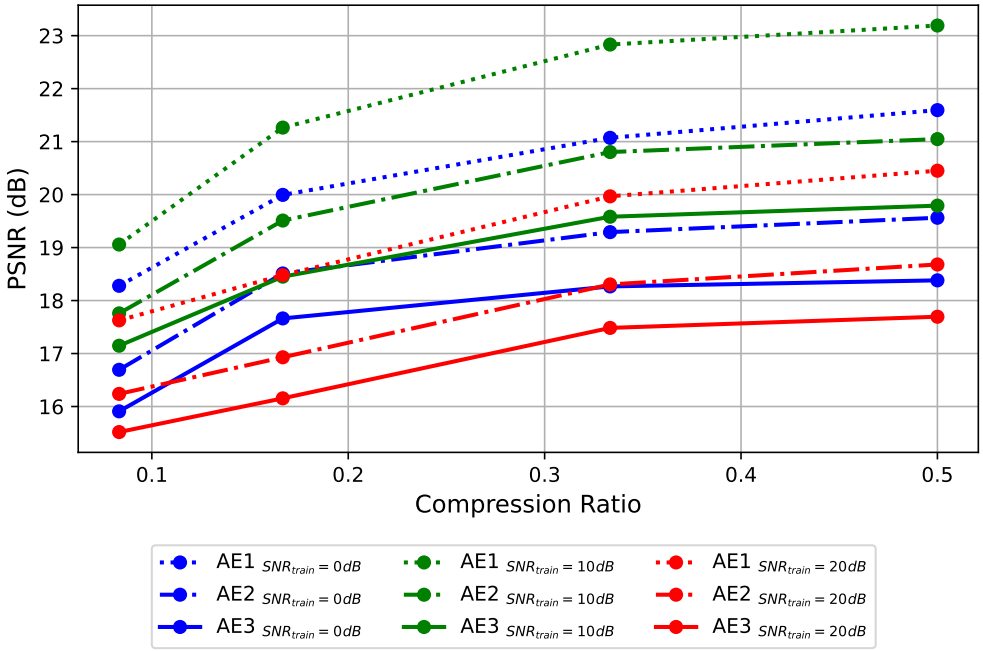

Figure 5.10: Performance of the trained 3-link DREC model with respect to compression ratio for different CSNR_{train} values (CSNR_{test}=CSNR_{train})

amplifier between each consecutive pair of channels, are considered. Since these layers are non-trainable, it is possible to load the parameters of the trained JCWR model to the 3-channels JCWR encoder and decoder. Then, the entire network is trained for an additional 50 epochs.

The performance of the two 3-link DREC model is presented in Figures 5.10 - 5.11. Figure 5.10 indicates, as expected, a decrease in performance after each node (decoding and encoding). However, for 3-link DREC, the degradation is small, suggesting that decoding and re-encoding the image helps mitigate the distortion caused by the channel. Figure 5.11 shows how the final performance changes for different combinations of CRs and CSNR_{train} values, compared to CSNR_{test}. It is clear that the decrease in performance from AE2 to AE3 is smaller than the decrease from AE1 to AE2. This indicates that the additional autoencoders act like denoising autoencoders.

The negligible reduction in performance leads to the conclusion that it might be possible to use the parameters from AE1 for all other nodes in a multi-link network without re-training each autoencoder for its position in the network. To investigate this scenario, a 5-link DREC model is simulated. The

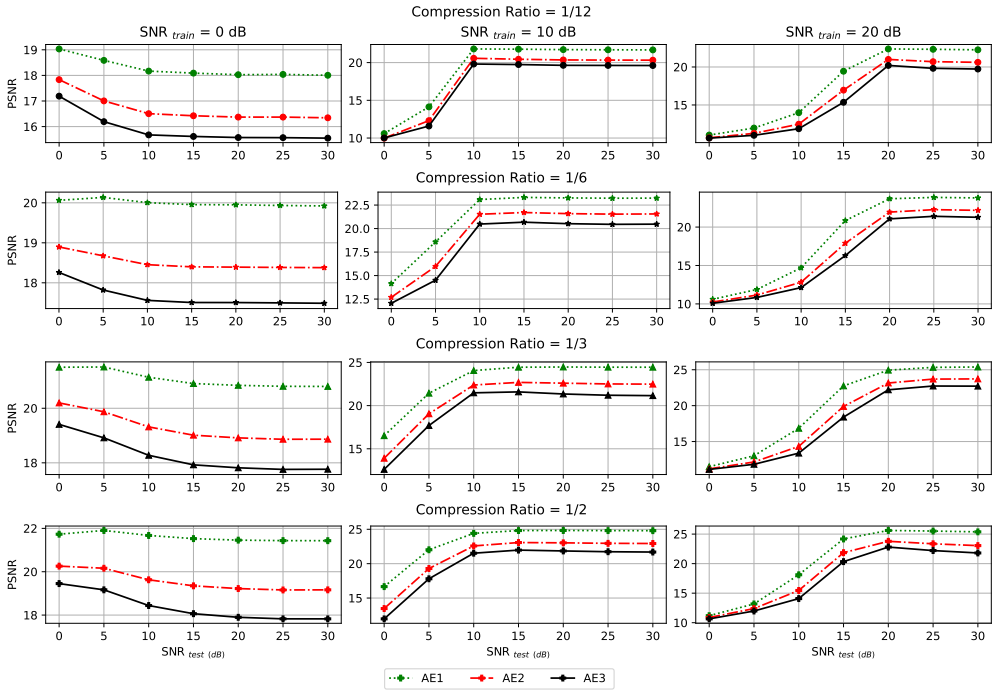

Figure 5.11: Performance of the trained 3-link DREC model for different CR and CSNR$_{test}$ values.

entire network structure is expanded in the same way as the 3-link expansion from the base DREC model. Parameters from AE1 are loaded into all five autoencoders without any extra training or tuning. Thus, all encoders are identical, and channels are considered the same as well. As illustrated in Figures 5.12 and 5.13, the difference between the final performance (AE5) and the first step (AE1) is around 2.5 dB for all cases. This means that by designing and training one optimal JSCC network (in this case, one autoencoder with one communication channel between encoder and decoder), it is possible to have a multi-link network with cascading the same nodes without requiring any extra actions like retraining or tuning.

The performance of the two 3-link JCWR model is presented in Figures 5.14 - 5.15. Figure 5.14 illustrates the cumulative effect of noise in the 3-channels JCWR. The result is in complete contrast to the JCWR model (5.5). In this model, amplification not only fails to improve the final performance but also significantly decreases it, especially for $CR \leq \frac{1}{3}$. Similar to the representation in 5.7, the transfer function of the combination of three channels with two amplifications can be obtained as shown in 5.8. It is evident that the noise

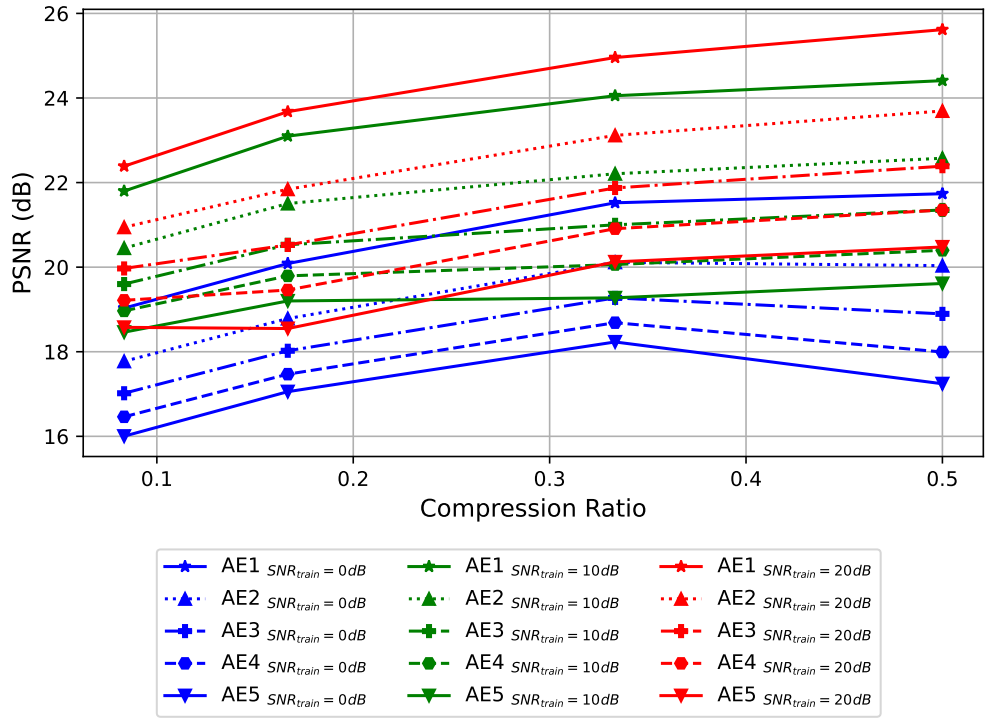

Figure 5.12: Performance of the 5-link DREC model with respect to compression ratio for different CSNR_{train} values (CSNR_{test}=CSNR_{train}).

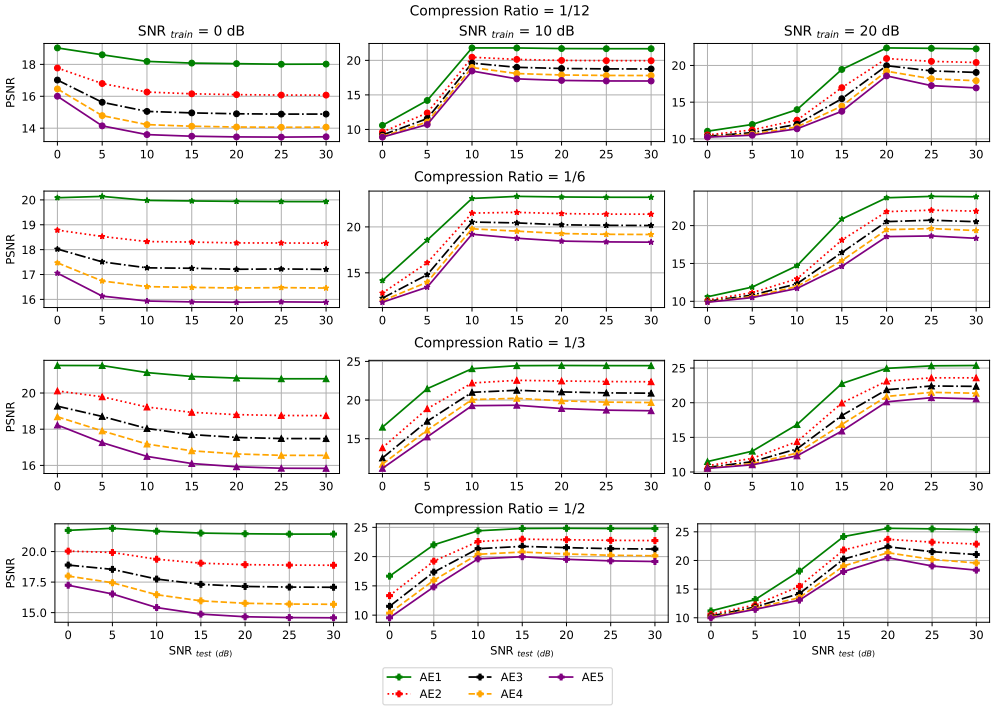

Figure 5.13: Performance of the 5-link DREC model for different CR and CSNR_{test} values.

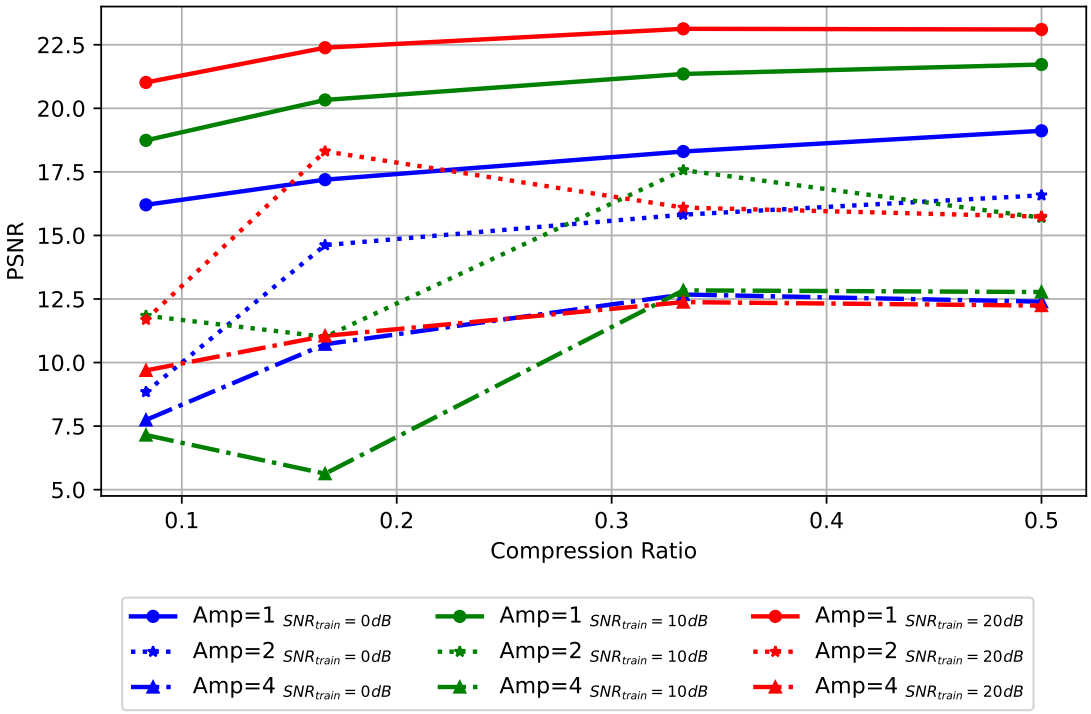

Figure 5.14: Performance of the trained 3-link JCWR model with respect to compression ratio for different CSNR_{train} values (CSNR_{test}=CSNR_{train})

increases rapidly with the growing number of channels and amplifiers. At higher compression levels, where all features of the image need to be encoded into a lower amount of symbols, distortions caused by this cumulative noise can hinder the entire system, ultimately affecting the decoder's ability to adapt.

$$\hat{z}_3 = h_3 z_1 + n_3 = (\alpha_2 \alpha_1 h_3 h_2 h_1) z_1 + (\alpha_2 \alpha_1 h_3 h_2 n_1 + \alpha_2 h_3 n_2 + n_3) \tag{5.8}$$

Figure 5.15 shows how amplification affects the performance for different combinations of CRs and CSNR_{train} values, compared to CSNR_{test}. It is clear that the performance gets worse as the amplification factor increases. This indicates that the amount of accumulated noise grows so much that it completely hides the original signal.

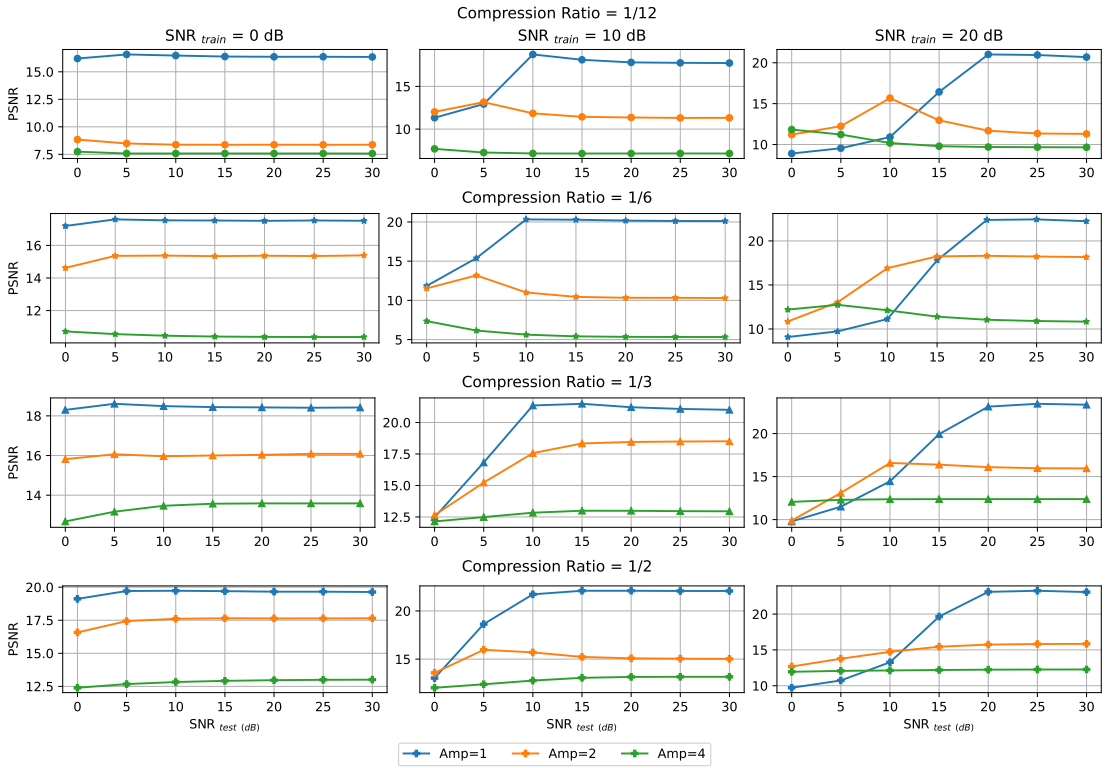

Figure 5.15: Performance of the trained 3-link JCWR model for different CR and CSNR$_{test}$ values.

6

Conclusions and Future Works

In conclusion, this research presented and discussed two models, namely the DREC and JCWR, designed for two-hop wireless communication networks. These models aim to address the challenges posed by noise and interference in wireless channels, particularly in scenarios involving multi-hop communication. The DREC model utilizes two autoencoders and two communication channels, demonstrating flexibility in adapting to different channel characteristics. On the other hand, the JCWR model employs a single autoencoder with an amplifier at the end of the first communication channel to enhance the signal before it undergoes the second channel.

The DREC model, with its dual autoencoders, allows for various configurations, providing adaptability to different channel conditions. It enables nodes to switch roles between sender, receiver, and intermediate nodes, allowing for efficient resource utilization. The model's core objective involves joint optimization, minimizing the average distortion between the original input image and its reconstruction. Security and privacy challenges arise due to potential unauthorized access and data leakage, which need careful consideration.

The JCWR model, employing a single autoencoder and an amplifier, presents an alternative approach. Each node can act as a sender, receiver, or amplifier, offering adaptability and flexibility in resource allocation. Security and privacy concerns are addressed by ensuring the amplifier only interacts with

encoded information, preserving data privacy. While the model's simplicity is advantageous, its effectiveness may be limited in scenarios with severe noise in the first communication channel.

The extension of both models to multi-link scenarios, accommodating three channel links, expands their applicability. The models' scalability and performance under various conditions need further exploration. In future works, the focus could shift towards enhancing security measures, mitigating latency introduced by double encoding and decoding processes, and investigating more sophisticated amplifiers for the JCWR model. Additionally, the models could benefit from optimization approaches such as reinforcement learning to dynamically adapt to changing channel conditions in real-time.

The evaluation of all models is done using the JSCC structure proposed in [1]. The study uses the COCO 2017 image dataset, with a subset of 50,000 training images and 10,000 test images. Models are developed using PyTorch, and evaluated using PSNR. Two main models, DREC and JCWR, are discussed, and their performance is analyzed under various conditions, including different compression ratios and signal-to-noise ratios.

For DREC, the impact of a second channel is explored, revealing a decrease in PSNR with higher compression and lower SNR. The model exhibits robustness to changes in channel quality. JCWR involves signal amplification between channels, and its performance is shown to vary based on the amplification factor and SNR. The cumulative noise effect in multi-channel systems is also examined.

The evaluation concludes by expanding these models to multiple links, indicating that additional channels in DREC have a minimal impact on performance, suggesting potential parameter sharing. However, JCWR with multiple links experiences a significant decrease in performance due to amplified noise accumulation.

The models' scalability and performance under various conditions need further exploration. In future works, the focus could shift towards considering security and privacy concepts, mitigating latency introduced by double encoding and decoding processes, reliability analysis of JSCC models, and investigating more sophisticated amplifiers for the JCWR model. Additionally,

the models could benefit from optimization approaches such as reinforcement learning to dynamically adapt to changing channel conditions in real-time.

Developing more sophisticated amplifiers for the JCWR model could enhance its performance, particularly in scenarios with severe noise in the first communication channel. Research could focus on adaptive amplification techniques that dynamically adjust the amplification factor based on channel conditions and noise levels.

Incorporating reinforcement learning algorithms into both models would enable them to dynamically adapt to changing channel conditions in real-time. This would allow the models to optimize their parameters and strategies based on the current network environment, leading to improved performance and efficiency.

A comprehensive analysis of the models' performance in multi-link scenarios is necessary to determine their scalability and effectiveness in more complex communication networks. This would involve evaluating the impact of additional channels on the models' ability to handle increased data transmission and varying network topologies.

References

[1] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz. "Deep Joint Source-Channel Coding for Wireless Image Transmission". In: *IEEE Transactions on Cognitive Communications and Networking* 5.3 (Sept. 2019), pp. 567–579. DOI: 10.1109/TCCN.2019.2919300.

[2] Zhengxue Cheng et al. *Deep Convolutional AutoEncoder-based Lossy Image Compression*. 2018. arXiv: 1804.09535 [cs.CV].

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[4] Deniz Gündüz and Elza Erkip. "Joint SourceChannel Codes for MIMO Block-Fading Channels". In: *IEEE Transactions on Information Theory* 54.1 (Jan. 2008). Student Member, IEEE and Senior Member, IEEE, pp. 333–342.

[5] Deniz Gündüz et al. "Source and Channel Coding for Correlated Sources over Multiuser Channels". In: *IEEE Transactions on Information Theory* 55.9 (Sept. 2009), pp. 3927–3944. DOI: 10.1109/TIT.2009.2025337.

[6] W. G. Hatcher and W. Yu. "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends". In: *IEEE Access* 6 (2018), pp. 24411–24432. DOI: 10.1109/ACCESS.2018.2830661.

[7] Mikolaj Jankowski, Deniz Gunduz, and Krystian Mikolajczyk. "Deep Joint Source-Channel Coding for Wireless Image Retrieval". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2020. DOI: 10.1109/icassp40776.2020.9054078. URL: https://doi.org/10.1109%2Ficassp40776.2020.9054078.

[8] D. B. Kurka and D. Gündüz. "DeepJSCC-f: Deep Joint Source-Channel Coding of Images With Feedback". In: *IEEE Journal on Selected Areas in Information Theory* 1.1 (May 2020), pp. 178–193. DOI: 10.1109/JSAIT.2020.2987203.

[9] David Burth Kurka and Deniz Gündüz. "Joint Source-Channel Coding of Images with (not very) Deep Learning". In: *International Zurich Seminar on Information and Communication (IZS 2020). Proceedings*. Feb. 2020, pp. 90–94. DOI: 10.3929/ethz-b-000402967.

[10] Changwoo Lee, Xiao Hu, and Hun-Seok Kim. *Deep Joint Source-Channel Coding with Iterative Source Error Correction*. 2023. arXiv: 2302.09174 [cs.LG].

[11] C. E. Shannon. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal* 27.3 (July 1948), pp. 379–423.

[12] Lucas Theis et al. *Lossy Image Compression with Compressive Autoencoders*. 2017. arXiv: 1703.00395 [stat.ML].

[13] George Toderici et al. "Full Resolution Image Compression with Recurrent Neural Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 5435–5443.

[14] J. Xu et al. "Wireless Image Transmission Using Deep Source Channel Coding With Attention Modules". In: *IEEE Transactions on Circuits and Systems for Video Technology* 32.4 (Apr. 2022), pp. 2315–2328. DOI: 10.1109/TCSVT.2021.3082521.

[15] Jialong Xu et al. *Deep Joint Source-Channel Coding for Semantic Communications*. 2023. arXiv: 2211.08747 [cs.IT].

[16] Fan Zhai, Yonatan Eisenberg, and Aggelos Katsaggelos. "Joint Source-Channel Coding for Video Communications". In: *Handbook of Image and Video Processing, Second Edition*. Elsevier, 2005, pp. 1065–1082. DOI: 10.1016/B978-012119792-6/50124-8.

Acknowledgments

I would like to express my deepest gratitude to my thesis advisor, Professor Federico Chiariotti, whose expertise, understanding, and patience, added considerably to my graduate experience. His profound knowledge and proficiency in various fields have not only enriched my academic journey but also provided invaluable guidance in my professional development. I am truly grateful for his mentorship and unwavering support.

I would also like to extend my sincere appreciation to Ph.D. student Francesco Pase for their insightful contributions to this thesis. Their constructive feedback and invaluable suggestions have significantly enhanced the quality of my work. I am truly grateful for their expertise and willingness to share their knowledge.