LEARNED TASK-AWARE COMPRESSION METHODS IN COMMUNICATION SYSTEMS

DISSERTATION

Submitted in Partial Fulfillment of

the Requirements for

the Degree of

DOCTOR OF PHILOSOPHY (Electrical Engineering)

at the

NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

by

Fabrizio Carpi

September 2024

LEARNED TASK-AWARE COMPRESSION METHODS IN COMMUNICATION SYSTEMS

DISSERTATION

Submitted in Partial Fulfillment of

the Requirements for

the Degree of

DOCTOR OF PHILOSOPHY (Electrical Engineering)

at the

NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

by

Fabrizio Carpi

September 2024

Approved:

Ivan Selesnick

Department Chair Signature

August 26, 2024

Date

Approved by the Guidance Committee:

Major: Electrical Engineering

Elza

Elza Erkip
Institute Professor
NYU Tandon School of Engineering
August 26, 2024
Date

Siddharth

Siddharth Garg
Institute Associate Professor
NYU Tandon School of Engineering
August 26, 2024
Date

SRangan

Sundeep Rangan
Professor
NYU Tandon School of Engineering
August 28, 2024
Date

S. S. Panwar

Shivendra Panwar
Professor
NYU Tandon School of Engineering
August 26, 2024
Date

Vita

Fabrizio Carpi received his Bachelor's Degree in Computer, Electronic and Communications Engineering and his Master's of Science in Communication Engineering from the University of Parma, Italy, in 2015 and 2018, respectively. During his Master's thesis in 2018, he was a visiting student at Duke University, NC. From 2018 to 2019, he was a research associate at the Internet of Things (IoT) Lab at University of Parma, Italy.

Since Fall 2019, he has been pursuing the Doctor of Philosophy degree in Electrical Engineering at New York University (NYU), Tandon School of Engineering, Brooklyn, NY, under the supervision of Prof. Elza Erkip and Prof. Siddharth Garg. He has also been a member of the NYU Wireless research center. His research interests include information theory, wireless communications, task-aware compression, and applied machine learning. During his doctoral studies, he received the Chang Education Award for teaching excellence in 2021, the Youla Award for research excellence in 2022, and the Goodman Award for leadership and academic excellence in 2024, all from NYU Tandon. He also received the Best Poster Award at the IEEE Communication Theory Workshop (CTW) in 2021, and the Best Student Paper Award at the IEEE Internation Workshop on Signal Processing Advances in Wireless Communications (SPAWC) in 2021.

Acknowledgements

I would like to express my deepest gratitude to my Ph.D. co-advisors, Prof. Elza Erkip and Prof. Siddharth Garg, for their invaluable guidance and all-around mentorship throughout this journey. I am truly fortunate to have benefited from their expertise and infectious enthusiasm, which inspired my academic path and broadened my future perspectives.

I extend my sincere thanks to my thesis committee members, Prof. Sundeep Rangan and Prof. Shivendra Panwar, for their precious feedback and advice during my time at New York University.

I am eternally grateful to my parents, Emilia and Pietro, my brother, Luca, and my sister-in-law, Chiara, for their unwavering support despite the distance that separates us. I thank them for their constant encouragement and sacrifices, and for giving me the strength to overcome challenges along the way.

I would like to thank my friends, who will forgive me for not naming them one-by-one because I fear leaving someone out, but will know I am referring to them. Thanks to my NYU colleagues and friends, for our shared memories and bonds created in these years. Thanks to my *Italian* friends in NYC, who quickly became a lifeline during the hardship of the pandemic years. Thanks to my *Campeginesi* friends in Italy, for always being by my side in this journey.

Saving the best for last, I want to express my heartfelt gratitude to Kara, who has brought boundless love and joy into my life. Her unwavering support and sacrifices have been an invaluable anchor. This milestone is just another starting point, and with her by my side, I know the best is yet to come.

Fabrizio

September 2024

To my parents, Emilia and Pietro, and my brother, Luca,
who always encouraged my dreams and supported me unconditionally.

To Kara, whose love and laughter are my constant inspiration.

ABSTRACT

LEARNED TASK-AWARE COMPRESSION METHODS IN COMMUNICATION SYSTEMS

by

Fabrizio Carpi

Co-Advisors: Prof. Elza Erkip, Ph.D, and Prof. Siddharth Garg, Ph.D.

Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy (Electrical Engineering)

September 2024

Traditionally, communication systems have been designed to optimize general performance metrics like error rate and signal distortion, regardless of the specific tasks performed over the networks. In contrast, semantic and task-aware communications aim to co-design a communication strategy tailored to individual applications, resulting in superior performance but also high specialization. In this thesis, we explore two communication settings within general-purpose networks, where we redefine the communication strategy by incorporating task-aware compression algorithms instead of classic separate processing blocks.

First, we introduce a precoding-oriented channel state information (CSI) feedback scheme for multi-cell multi-user MIMO systems. Our learned end-to-end architecture integrates the downlink channel estimation phase, the CSI compression, and the downlink precoding computation. The proposed loss function maximizes the users' achievable rates while minimizing the CSI feedback overhead. Simulations demonstrate superior performance compared to previous precoding-oriented methods, and increased efficiency compared to conventional methods that separate the CSI compression blocks from the precoding processing.

Second, we consider the primitive relay channel, which consists of a source-destination pair along with a relay, where we propose detection-oriented compress-and-forward (CF) neural relays. By training neural CF relays with decoder-side information, we maximize the end-to-end communication rate between the source and the destination. Our learned compressor recovers binning of the quantized indices at the relay, mimicking the optimal asymptotic CF strategy without requiring prior knowledge imposed in the design. We show the advantages of exploiting the correlated destination signal for relay compression through different neural CF architectures. Our learned task-oriented compressors provide the first proof-of-concept work toward interpretable and practical neural CF relaying schemes.

Table of Contents

List of Figures

Chapter 1

Introduction

In his seminal work [6], Shannon defined the problem of communication as the accurate reproduction of a message at a remote location. In this spirit, he intentionally separated the *semantic aspect* from the *engineering problem*, representing information with a set of possible messages characterized by a probability distribution. This led to the establishment of classical information theory, focusing on task-agnostic measures like entropy and mutual information. Shannon and Weaver [7] further refined this framework, identifying three levels in the analysis of communication problems:

1. the *technical* problem (how accurately can symbols be transmitted?),
2. the *semantic* problem (how precisely do transmitted symbols convey meaning?),
3. the *effectiveness* problem (how effectively does the received meaning affect conduct?).

However, their primary focus remained on the technical aspect and in particular on how to overcome noisy distortions in the communication channel. This emphasis on the technical aspects of communication led to the development of numerous technologies optimized for metrics like data rate and error probability, often overlooking the specific tasks and applications that the communication system was planned to support.

Fast-forward several decades, and the advent of machine learning (ML) and artificial intelligence (AI) is revolutionizing the field of communication systems. Building upon Shannon's foundational work on the technical problem, ML/AI is now being integrated as an optimization tool to solve open problems using a data-driven approach. Traditionally, communication systems relied on algorithms and protocols designed for precise assumptions and channel models. However, the integration of ML/AI has introduced a new paradigm, enabling communication systems to learn, adapt, and optimize their performance in complex environments in an end-to-end fashion [8]. This has led to significant advancements in areas such as channel estimation and signal detection [9, 10], resource allocation, and interference management [11, 12]. By identifying use cases where traditional algorithms fall short, ML/AI is not only improving existing systems but also opening up new possibilities for network optimization across a wide range of performance metrics.

Concurrently, going back to Shannon and Weaver [7], recent years have seen growing interest in the second and third levels of problems, i.e., the semantic and the effectiveness problems. This paradigm shift, known as task-aware (or semantic) communication design, recognizes that messages are not only sequences of bits, but conveyors of diverse concepts with varying importance depending on the final goal. In this context, a semantic/effective transmitter prioritizes the transmission of information relevant to the specific task at hand. As an example, consider an image classification task (dog vs cat): instead of transmitting the whole sequence of pixels, a semantic/effective transmitter would send relevant features such as the shape of the ears, rather than irrelevant information like the image background.

ML/AI has played a pivotal role in enabling the recent explosion of task-aware (semantic) communications, see [13, 14, 15, 16, 17, 18] for an introduction and references therein. The ability to extract meaningful patterns from data has facilitated the development of tailored algorithms that can adapt communication strategies to specific tasks. From the wireless industry perspective, 6G is expected to provide connectivity to emerging technologies such as autonomous robots (including vehicles), digital twins, augmented/virtual reality, and personal assistants (based on generative AI) running on the cloud. The diverse nature of these use cases may necessitate highly specialized network architectures to deliver the required performance levels, underscoring the importance of continued research and development in task-aware communication design.

Boosted by the capabilities of ML/AI tools and inspired by recent advancements in task-aware communications, this thesis shifts focus from emerging use cases to identifying intrinsic tasks within existing networks. We build upon the task-aware foundation and redefine processing blocks that are used in general-purpose networks. Our strategy involves a systematic approach:

1. identifying specific tasks within the network;
2. defining key metrics and objectives for the optimization;
3. co-designing the communication strategy in conjunction with the identified task.

This is particularly relevant as the demand for efficient and adaptable communication systems grows. By redefining traditional processing blocks in a task-aware fashion, we aim to improve overall performance and efficiency, ensuring that general-purpose networks can effectively support the increasingly diverse demands of modern applications.

In this thesis, we specifically focus on the task-aware compression perspective. For a given application, our goal is to identify the appropriate metrics and design variables to optimize the communication system for that particular task. Our focus is on developing strategies that describe the source messages with the minimum amount of overhead while maximizing the end-to-end performance metrics. To this end, we investigate two task-aware communication problems.

- The channel state information (CSI) feedback problem. In this scenario, we consider downlink beamforming as the ultimate task for the base station. We adopt a joint processing solution that bypasses explicit CSI reconstruction, allowing the base station to directly derive beamforming information from the compressed CSI. This approach enables us to design a CSI compression scheme that maximizes users' achievable rates in an end-to-end fashion.

- The compress-and-forward (CF) problem in the primitive relay channel. In this scenario, the goal is to detect the transmitted source symbols at the destination, with the help of a relay node. The relay's goal is to design a CF strategy that maximizes the source-to-destination communication rate. In this case, we exploit the distributed compression architecture and design a CF strategy that exploits the correlated destination signal for the compression.

In both scenarios, we leverage the task-aware approach to design specialized mechanisms that can be smoothly integrated into general-purpose networks. Specifically, we use neural networks to model the relevant processing blocks of the communication system. We then define an end-to-end loss function that directly optimizes the task-aware objective. This approach empowers the learned processing blocks to extract and process task-relevant features effectively. Finally, we provide an interpretation of the obtained results to shed light on the inner workings of the "black box" view of neural networks, improving our understanding of the underlying learned mechanisms.

1.1 Thesis Organization

The rest of this thesis is organized as follows. An overview of general tools for task-aware compression is given in Chapter 2. The precoding-oriented CSI feedback problem is discussed in Chapter 3. The detection-oriented relays are analyzed in Chapter 4. Application-specific conclusions are drawn at the end of both Chapter 3 and Chapter 4, while more comprehensive takeaways messages are discussed in Chapter 5.

Chapter 2

Task-Aware Compression

The key ingredient of task-aware compression is to use compression methods that can be co-designed with the task of interest. In particular, we seek to minimize the communication overhead (or compression rate) while maximizing some task-related utilities. While identifying the utilities is often straightforward, as they are directly linked to performance metrics, managing the communication overhead can be problematic. In many compression problems, the optimization of the compressor becomes intractable when the quantization is performed in high-dimensional spaces [19]. In this chapter, we first present a simple example of task-aware compression design, focusing on optimizing the compression for the hypothesis testing problem. This simplified setting illustrates how the task utility and communication strategy can be co-designed. We then move to define the important concept of transform coding [2], which allows to further expand practical aspects of the task-aware communication co-design. Finally, we summarize recent advances in neural compression, introducing tools that will be used in subsequent chapters for the design of end-to-end task-aware compression methods for communication systems.

The rest of this chapter is organized as follows. Section 2.1 discusses the compression for hypothesis testing example. Section 2.2 introduces the concept of transform coding. Section 2.3 explains recent advancements in neural compression.

2.1 Example: Compression for Hypothesis Testing

This section analyzes a simple task-aware compression design. The focus is on the analytical construction of compressors for the binary hypothesis testing problem. The setup is similar to the one of distributed detection [21]. In this work, we focus on the algorithmic analysis of the task-aware compressor.

We consider a binary hypothesis testing scenario where the resource-constrained client (transmitter) performs fixed-length single-shot compression on data sampled from one of two distributions; the server (receiver) performs a hypothesis test on multiple received samples to determine the correct source distribution. To this end, the task-aware compression problem is formulated as finding the optimal source coder that maximizes the asymptotic error performance of the hypothesis test on the server side under a rate constraint. A new source coding strategy based on a greedy optimization procedure is proposed and it is shown that the proposed compression scheme outperforms universal fixed-length single-shot coding scheme for a range of rate constraints.

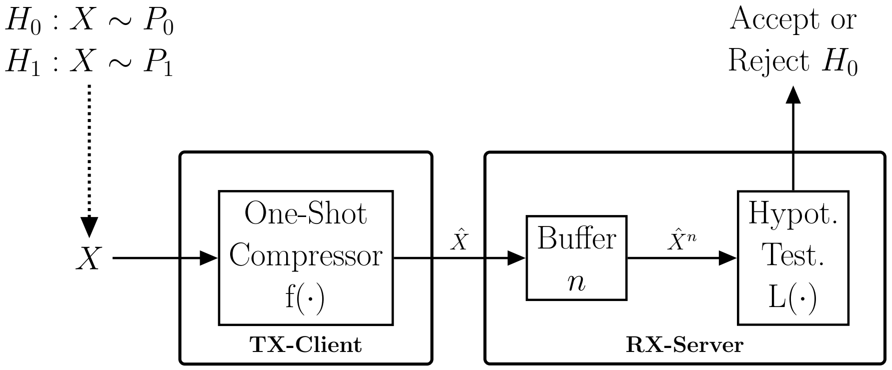


Figure 2.1: System model: binary hypothesis testing.

2.1.1 Binary Hypothesis Testing under Single-Shot Compression

Consider the system model of Fig. 2.1. The source data comes from one of the two distributions P_θ, $\theta \in \{0, 1\}$, where $\theta = 0$ represents the null hypothesis H_0 and $\theta = 1$ represents the alternative hypothesis H_1. We have $X_1, \dots, X_n \sim P_\theta$ i.i.d. random variables defined over a finite alphabet $\mathcal{X} = \{1, \dots, |\mathcal{X}|\}$. The transmitter, due to memory constraints, cannot store and process X^n jointly to do hypothesis testing. Instead, it sends the one-shot (scalar) compressed X^n to the receiver where hypothesis testing takes place. Note that this setup is equivalent to the distributed detection problem [21], where n transmitters send i.i.d. samples to a central unit to perform the hypothesis testing (detection).

At the transmitter, the single-shot compressor f is a surjective function defined as

$$\mathrm{f} : \mathcal{X} \to \mathcal{M} \tag{2.1}$$

where $\mathcal{M} = \{1, \dots, M\}$ is the compressed alphabet of size M. We denote $\hat{X} = \mathrm{f}(X)$,

Part of this work was presented at IEEE SPAWC 2021 [20].

i.e., $\hat{X}$ represents the mapping of the source letter X. We consider $M < |\mathcal{X}|$, since for $M \geq |\mathcal{X}|$ there is no need for compression. This corresponds to fixed-rate compression with rate $R = \log M$.[1]

The probability distribution of $\hat{X}$ under P_θ, $\theta \in \{0,1\}$, is denoted as $\hat{P}_\theta$ and is given by

$$\hat{P}_\theta(\hat{x}) = \sum_{x:\mathrm{f}(x)=\hat{x}} P_\theta(x). \tag{2.2}$$

The receiver observes $\hat{X}_1, \dots, \hat{X}_n$ and either accepts or rejects the null hypothesis. Using standards definitions in simple hypothesis testing [22], type-I error, denoted as α_n, occurs when the null hypothesis ($\theta = 0$) is true, but the receiver rejects it. Instead, type-II error, denoted as β_n, corresponds to the receiver accepting the null hypothesis when the alternative hypothesis ($\theta = 1$) is true. It is known that in the classical hypothesis testing setting, for any $\epsilon \in (0, 1/2)$ and $\alpha_n < \epsilon$, the optimal type-II error β_n^ϵ decays exponentially in n with exponent γ defined as

$$\gamma = -\lim_{n\to\infty} \frac{1}{n} \log \beta_n^\epsilon. \tag{2.3}$$

We say that (R, η) is *achievable* if there exists a single-shot rate R compressor at the client and a corresponding hypothesis testing function at the server with type-I error less than ϵ and type-II error exponent η. Note that type-II error exponent does not typically depend on type-I error bound ϵ [22] as long as ϵ is fixed, hence we will not explicitly state the dependency on ϵ. In particular, for a given compression rate R, we would like to find the largest achievable type-II error exponent

$$\gamma^\star(R) = \sup\{\eta : (R, \eta) \text{ achievable}\}. \tag{2.4}$$

[1] Throughout this thesis $\log(\cdot)$ is assumed to be base 2.

Note that if $R = \log(|\mathcal{X}|)$ and the compressor is the identity transformation $\mathrm{id}(\cdot)$, then Chernoff-Stein lemma [22] determines the optimal error exponent

$$\gamma^\star(\log|\mathcal{X}|) = \gamma_{\mathrm{id}}(\log|\mathcal{X}|) = D(P_0||P_1), \tag{2.5}$$

where $D(P_0||P_1)$ is the Kullback–Leibler (KL) divergence between P_0 and P_1 [22]. The error exponent penalty for a rate R compressor f at is defined as

$$\Delta_{\mathrm{f}}(R) = D(P_0||P_1) - \gamma_{\mathrm{f}}(R), \tag{2.6}$$

where $\gamma_{\mathrm{f}}(R)$ is the largest type-II error exponent determined by the compressor f. The optimal penalty is

$$\Delta^\star(R) = D(P_0||P_1) - \gamma^\star(R). \tag{2.7}$$

Since the hypothesis testing is eventually performed on the compressed variable $\hat{X}$, we need to establish optimality of the log-likelihood ratio (LLR) test.

Lemma 1 (Hypothesis testing on compressed variables)**.** *The following LLR test on compressed variables $\hat{X}_i = \mathrm{f}(X_i)$, $i = 1, \ldots, n$, is optimal.*

$$L(\hat{X}_1, \ldots, \hat{X}_n) = \sum_{i=1}^{n} \log \frac{\hat{P}_0(\hat{X}_i)}{\hat{P}_1(\hat{X}_i)} \underset{\hat{\theta}=1}{\overset{\hat{\theta}=0}{\gtrless}} \log T, \tag{2.8}$$

where $T \geq 0$ depends on the type-I error exponent bound ϵ. The corresponding optimal error exponent is

$$\gamma_{\mathrm{f}}(R) = D(\hat{P}_0||\hat{P}_1). \tag{2.9}$$

Proof sketch. Since the source random variable is i.i.d. and the compressor function is f memoryless, the compressed variable is also i.i.d. $\hat{X}_1, \ldots, \hat{X}_n \sim \hat{P}_\theta$. Then, Neyman-Pearson test [22, Chapter 11] can be applied to $\hat{X}^n$. Moreover, Chernoff-Stein lemma determines that the the optimal error exponent is equal to the KL divergence between the distribution of the compressed variables under the two hypotheses. □

Note the error exponent $\gamma_{\mathrm{f}}(R)$ determines the speed of convergence — intuitively, the farther apart the two compressed distributions (large KL divergence), the faster the type-II error probability goes to zero. Hence, our goal is to find a compressor f which induces a partition of M sets over $\mathcal{X}$ such that the KL distance between the compressed distributions $D(\hat{P}_0||\hat{P}_1)$ is maximized. Clearly, compression reduces the error exponent (we will mathematically show this in Proposition 1) and by Lemma 1 the smallest compression penalty for the compressor f is

$$\Delta_{\mathrm{f}}(R) = D(P_0||P_1) - D(\hat{P}_0||\hat{P}_1). \tag{2.10}$$

Then, the optimal compressor $\mathrm{f}^\star$ at rate $R = \log M$ is

$$\mathrm{f}^\star = \arg\max_{\mathrm{f}} D(\hat{P}_0||\hat{P}_1) \quad \text{s.t. } |\mathrm{f}| \leq M, \tag{2.11}$$

or, equivalently,

$$\mathrm{f}^\star = \arg\min_{\mathrm{f}} \Delta_{\mathrm{f}}(R) \quad \text{s.t. } |\mathrm{f}| \leq M. \tag{2.12}$$

where $|\mathrm{f}|$ is the cardinality of the compression function.

In the following proposition, we derive a useful analytical expression for $\Delta_{\mathrm{f}}(R)$ in terms of distributions over compressed symbols. For mathematical convenience, we define $\mathcal{G}_{\hat{x}} = \{x : \mathrm{f}(x) = \hat{x}\}$; this set includes the source outcomes mapped to the compressed symbol $\hat{x}$. Hence, the compressor induces the "groups" $\mathcal{G}_{\hat{x}} \in \{\mathcal{G}_1, \ldots, \mathcal{G}_M\} = \mathcal{G}$, which form a partition over $\mathcal{X}$.

Proposition 1 (Compression penalty on type-II error exponent)**.** *For any compressor* f*, the minimal compression penalty is* $\Delta_{\mathrm{f}}(R) \geq 0$ *and can be expressed as:*

$$\Delta_{\mathrm{f}}(R) = \sum_{\hat{x}=1}^{M} \hat{P}_0(\hat{x})\, D\Big(P_0(x|\hat{x})\Big|\Big|P_1(x|\hat{x})\Big) \tag{2.13}$$

where the posterior distribution of X given the compressed realization $\mathrm{f}(X) = \hat{x}$ *is*

$$P_\theta(x|\hat{x}) = \frac{P_\theta(x)}{\hat{P}_\theta(\hat{x})} \mathbb{1}\{\hat{x} = \mathrm{f}(x)\}. \tag{2.14}$$

Proof. Expanding equation (2.10):

$$\begin{aligned}
\Delta_{\mathrm{f}}(R) &= \sum_{x\in\mathcal{X}} P_0(x) \log \frac{P_0(x)}{P_1(x)} - \sum_{\hat{x}\in\mathcal{M}} \hat{P}_0(\hat{x}) \log \frac{\hat{P}_0(\hat{x})}{\hat{P}_1(\hat{x})} \\
&= \sum_{\hat{x}\in\mathcal{M}} \sum_{x\in\mathcal{G}_{\hat{x}}} P_0(x) \log \frac{P_0(x)}{P_1(x)} - \sum_{\hat{x}\in\mathcal{M}} \left(\sum_{x\in\mathcal{G}_{\hat{x}}} P_0(x) \right) \log \frac{\hat{P}_0(\hat{x})}{\hat{P}_1(\hat{x})} && (2.15) \\
&= \sum_{\hat{x}\in\mathcal{M}} \sum_{x\in\mathcal{G}_{\hat{x}}} P_0(x) \log \left(\frac{P_0(x)}{\hat{P}_0(\hat{x})} \frac{\hat{P}_1(\hat{x})}{P_1(x)} \right) && (2.16) \\
&= \sum_{\hat{x}\in\mathcal{M}} \sum_{x\in\mathcal{G}_{\hat{x}}} P_0(x) \log \frac{P_0(x|\hat{x})}{P_1(x|\hat{x})} && (2.17) \\
&= \sum_{\hat{x}\in\mathcal{M}} \hat{P}_0(\hat{x})\, D\Big(P_0(x|\hat{x})\Big|\Big|P_1(x|\hat{x})\Big)
\end{aligned}$$

where: in (2.15) we used the definition (2.2); in (2.15) and (2.16) we used the fact that $\mathcal{G}_1, \ldots, \mathcal{G}_M$ form a partition over $\mathcal{X}$; in (2.17) we used the definition (2.14) since $P(\hat{X}|X) = \mathbb{1}\{\hat{X} = \mathrm{f}(X)\}$. Note that if $\mathcal{G}_{\hat{x}}$ contains a single element (one-to-one mapping), then $D\big(P_0(x|\hat{x})||P_1(x|\hat{x})\big) = 0$. Moreover, (2.15) is greater than zero by the log-sum inequality. □

Algorithm 1: KL-greedy compressor's construction

```
Input   : Source distributions $P_0, P_1$; rate $M$.
Initialize: $\hat{P}_0 \leftarrow P_0$, $\hat{P}_1 \leftarrow P_1$, $\mathcal{G} \leftarrow \{\{1\},\dots,\{|\mathcal{X}|\}\}$.
for $k = 1,\dots,|\mathcal{X}| - M$ do
    Find $\{\mathcal{G}_a, \mathcal{G}_b\} \subset \mathcal{M}_k$ which minimize (2.18).
    Remove the $b$-th entry and combine $\{\mathcal{G}_a, \mathcal{G}_b\}$ by updating the $a$-th entry:
    $\hat{P}_0 \leftarrow [\dots, \hat{P}_0(\mathcal{G}_a) + \hat{P}_0(\mathcal{G}_b), \dots, 0, \dots]$
    $\hat{P}_1 \leftarrow [\dots, \hat{P}_1(\mathcal{G}_a) + \hat{P}_1(\mathcal{G}_b), \dots, 0, \dots]$
    $\mathcal{G} \leftarrow [\dots, \mathcal{G}_a \cup \mathcal{G}_b, \dots, \emptyset, \dots]$
end
Output : Compressed distr. $\hat{P}_0, \hat{P}_1$; groups $\mathcal{G}$.
```

Non-negativity of $\Delta_{\mathrm{f}}(R) \geq 0$ can also be observed from equation (2.13) as it is a convex combination of KL-distances, each individually positive. Proposition 1 also yields an important intuition about optimal compression: note that the $\hat{x}$-th term in (2.13) is directly proportional to the relative entropy between the posteriors over the $\hat{x}$-th group $\mathcal{G}_{\hat{x}}$ induced by f. As a consequence, (2.13) suggests that a good task-aware compression strategy combines the source letters that have similar posteriors over the compressed groups; in other words, the probability ratios between the combined letters under P_0 have to be similar to the ones under P_1.

2.1.2 Proposed Compressor for Hypothesis Testing

When solving the optimization problem in (2.11), one has to consider all the possible surjective functions f which induce valid partitions over the source alphabet; the number of such partitions is exponential in the source/compressed alphabet size. Partitioning problems of this nature have been shown to be NP-Hard [23, Chapter 3],[24]. Similar considerations were drawn in [21], where it was concluded that an exhaustive search is needed.

Previous work [21] focused on the special case of one-bit quantization ($M = 2$), where a simple threshold rule provides the optimal compressor. On the other hand, in this thesis, we propose an efficient (i.e., polynomial time) greedy approximation for the optimal compressor for any M.

The following lemma is the basis for our construction.

Lemma 2 (One-step Compression from $|\mathcal{X}|$ to $|\mathcal{X}|-1$)**.** *Let* f *be a compression rule that groups two letters $\{a,b\} \subset \mathcal{X}$. That is, $\mathcal{G}_m = \{a,b\}$, $m \in \mathcal{M}$, and the others groups $\mathcal{G}_i$, $i = 1,\dots,M$, $i \neq m$, are one-to-one. Then, the optimal compressor for $M = |\mathcal{X}| - 1$ induces the groups $\mathcal{G}^\star$, minimizing the compression penalty*

$$\mathcal{G}^\star = \underset{\mathcal{G}_m=\{a,b\}\subset\mathcal{X}}{\arg\min} \left\{ \hat{P}_0(m) D\Big(P_0(x|m) \Big|\Big| P_1(x|m) \Big) \right\}, \tag{2.18}$$

where the posteriors over the candidate group $\mathcal{G}_m = \{a,b\}$ are simply defined as

$$P_\theta(x|m) = \left[\frac{P_\theta(a)}{P_\theta(a)+P_\theta(b)}, \frac{P_\theta(b)}{P_\theta(a)+P_\theta(b)} \right]. \tag{2.19}$$

Note that if the groups $\mathcal{G}_i$ are one-to-one, the i-th KL divergence term in (2.13) is 0. Intuitively, when reducing the alphabet size by one, the optimal compressor combines the two letters that minimize the product of the probability of the group and the KL distance between the posteriors over the group.

For general M, we propose an iterative construction of the compressor that reduces the compressed alphabet size by one in each step. Denote the steps by $k = 1,\dots,|\mathcal{X}| - M$, where M is the target rate. Let $\mathcal{M}_k$ be the compressed alphabet at the k-th step, with size $|\mathcal{M}_k| = |\mathcal{X}| - k$, with $k = 1,\dots,|\mathcal{X}| - M$. Let $\mathcal{G}_1,\dots,\mathcal{G}_{|\mathcal{M}_k|}$ be the corresponding partition on $\mathcal{X}$ at the k-th step. For example, at the first step $k = 1$, the (optimal) groups $\mathcal{G}_1,\dots,\mathcal{G}_{|\mathcal{X}|-1}$ are computed according to Lemma 2. Generally, at step $k > 1$, our compressor combines the two groups

$\{\mathcal{G}_a, \mathcal{G}_b\}_k^\star \subset \mathcal{M}_k$ that minimize (2.18), where $\mathcal{X}$ is replaced by $\mathcal{M}_k$ and $\{\mathcal{G}_a, \mathcal{G}_b\}$ is a generalization of $\{a, b\}$. Finally, the compression function f is defined such that $\mathrm{f}(x) = \hat{x}$ if $x \in \mathcal{G}_{\hat{x}}$. We call our proposed compressor "KL-greedy" and its construction is summarized in Algorithm 1. Note that the number of pairs of groups $\{\mathcal{G}_a, \mathcal{G}_b\}$ that need to be considered at the k-th step is $\binom{|\mathcal{M}_k|}{2}$. Thus, our compressor can be designed in polynomial time.

2.1.3 Compressed Hypothesis Testing Results

In this section, we discuss numerical results and performance of Algorithm 1.

For illustration purposes, we consider P_θ to be a (shifted) binomial distribution over $\mathcal{X}$ with parameter s_θ, i.e.,

$$P_\theta(x) = \binom{|\mathcal{X}| - 1}{x - 1} s_\theta^{x-1} (1 - s_\theta)^{|\mathcal{X}|-x}. \tag{2.20}$$

We quantify the compression penalty $\Delta_{\mathrm{f}}(R)$ based on (2.10). We also estimate type-II error rate by performing the LLR test (2.8) on the receiver side; we consider blocklength $n = 5$ and bound on the type-I error $\epsilon = 0.05$. The threshold T is empirically chosen such that it is the largest value for which the estimated type-I error is $N(\hat{\theta} = 1, \theta = 0)/N(\theta = 0) < \epsilon$, for a given compressor f at rate M; $N(\cdot)$ is the counting function. The type-II error rate is empirically estimated as $N(\hat{\theta} = 0, \theta = 1)/N(\theta = 1)$. Both estimates are computed over $N(\theta = 0) = N(\theta = 1) = 10^6$ realizations of source blocks x^n.

As a baseline, we consider the universal fixed-length single-shot lossy compression scheme analyzed in [1]. We recall that although this universal compressor is task-unaware, it is designed for soft reconstruction under logarithmic loss distortion, which generally provides "universally good" schemes [25]. The construction of this universal compressor aims to find a universal distribution over $\mathcal{X}$ which is used to approximate the source distribution over the family $\{P_0, P_1\}$. Intuitively, the most likely letters of the universal distribution get one-to-one mappings, while the least likely ones get grouped together.

In the figures, we show empirical results for different compression functions f:

- Uncompressed: no compression is performed, i.e., $\hat{x} = x$;

- Optimal compressor: defined in (2.12);

- Our KL-greedy compressor: defined in Section 2.1.2 and Algorithm 1;

- Universal compressor: defined in [1].

In Fig. 2.2, 2.3 and 2.4 we consider a source alphabet of size $|\mathcal{X}| = 13$; the parameters of the two hypotheses are $s_0 = 0.4$, $s_1 = 0.6$. On the other hand, in Fig. 2.5 and 2.6 we consider a larger source alphabet of size $|\mathcal{X}| = 256$; the parameters of the two hypotheses are $s_0 = 0.48$, $s_1 = 0.52$. We note that for this larger source alphabet, it is no longer computationally feasible to determine the optimal compressor.

Fig. 2.2 illustrates the resulting KL-greedy compressor, the universal compressor, and the compressed distributions for $M = 4$. As previously explained, our KL-greedy compressor seeks to maximize the KL distance between the posteriors over the groups; we also point out that this induces a partition on $\mathcal{X}$ that divides the source alphabet into regions where one of the hypotheses is more likely than the other. This pattern is also visible in the compressed distributions since the two hypotheses exhibit divergent distributions (large KL distance). On the other hand,

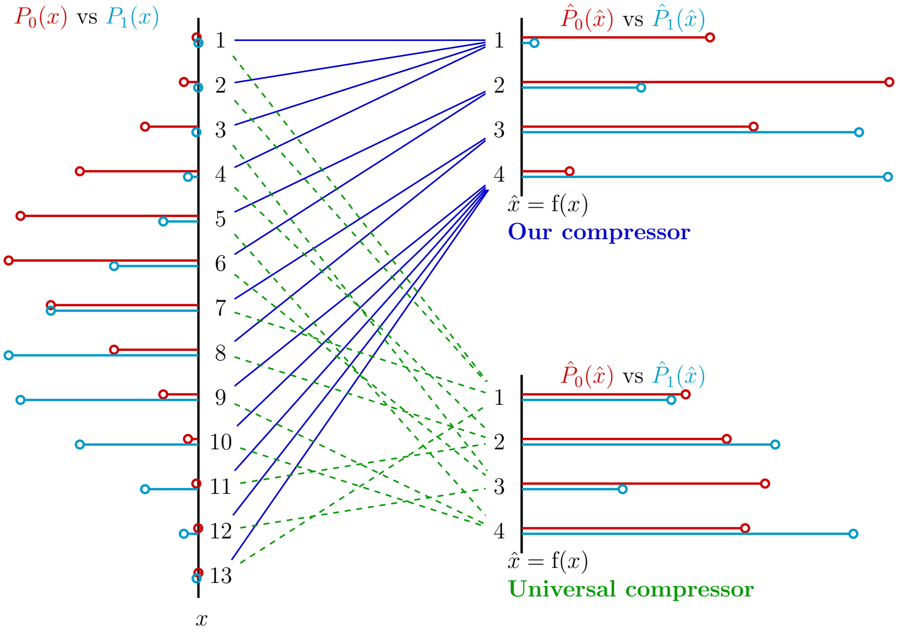

$P_0(x)$ vs $P_1(x)$
x
1
2
3
4
5
6
7
8
9
10
11
12
13
$\hat{P}_0(\hat{x})$ vs $\hat{P}_1(\hat{x})$
$\hat{x} = \mathrm{f}(x)$
Our compressor
$\hat{P}_0(\hat{x})$ vs $\hat{P}_1(\hat{x})$
$\hat{x} = \mathrm{f}(x)$
Universal compressor

Figure 2.2: Left: Source distributions for $|\mathcal{X}| = 13$. Top-right: compressed distributions for our compressor of Algorithm 1; the solid blue line shows the mappings of the compression function. Bottom-right: compressed distributions for the universal compressor from [1]; the dashed green line shows the mappings of the compression function.

the universal compressor aims to make the two compressed distributions as uniform as possible. As we discussed above, the larger the KL divergence between the compressed distributions, the better for the hypothesis testing task.

Fig. 2.3 and 2.5 show the compression penalty as a function of the compression rate M. The former also shows the performance of the optimal compressor, since it can be computed in reasonable time for a small source alphabet; in this case, we can see that our compressor performs close to the optimal. In both cases, our compressor outperforms the universal compressor, and it quickly achieves zero penalty, i.e., the KL distance of the compressed distributions is close to the uncompressed one as M increases.

Fig. 2.4 and 2.6 show the empirical type-II error rate as a function of the

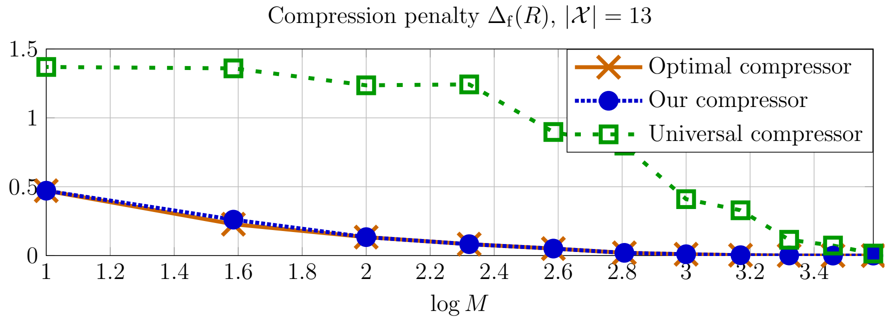


Figure 2.3: Compression penalty for $|\mathcal{X}| = 13$.

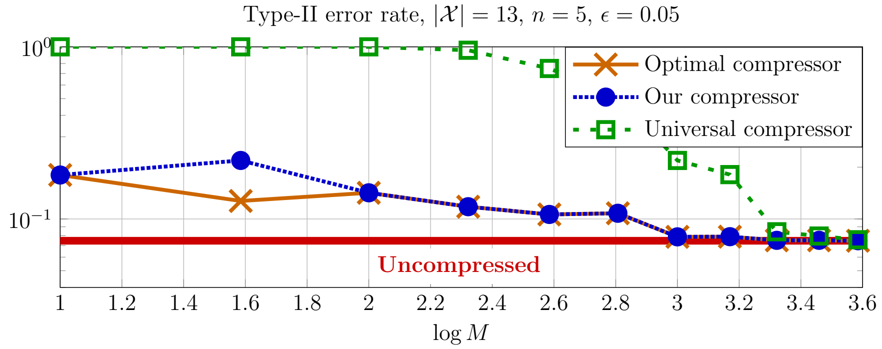


Figure 2.4: Type-II error rates for $|\mathcal{X}| = 13$.

compression rate M. The former also shows the performance of the optimal compressor: our compressor performance overlaps with the optimal compressor. For both the small and the large alphabet scenarios, our compressor outperforms the universal compressor, and it quickly achieves an error rate close to the uncompressed setting as M increases.

2.2 Transform Coding

Shifting our focus, we now introduce the broader concept of transform coding, a widely adopted technique in signal processing, including applications in image, audio, and video compression. The fundamental idea behind transform coding is

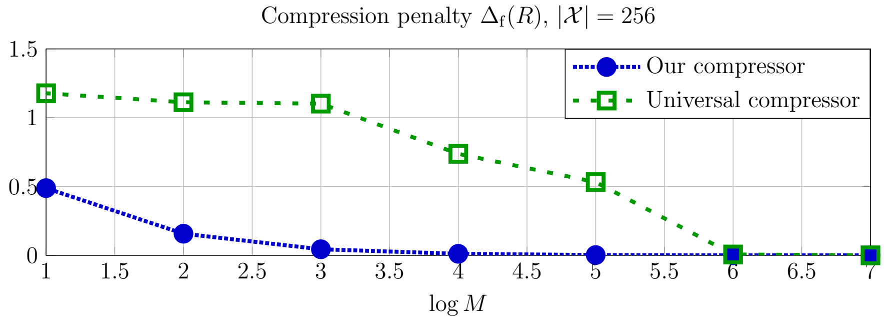


Figure 2.5: Compression penalty for $|\mathcal{X}| = 256$.

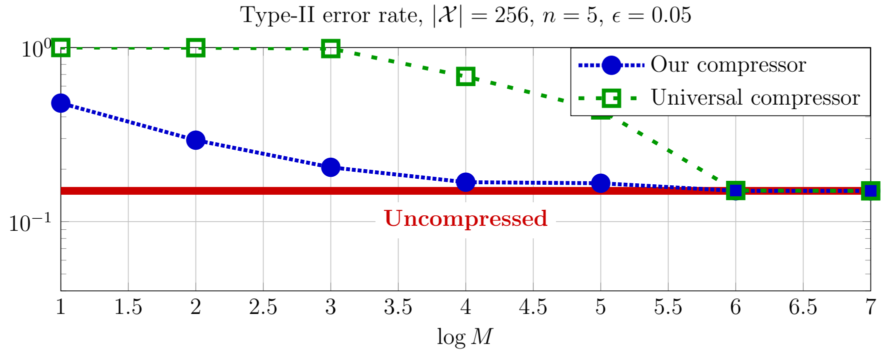


Figure 2.6: Type-II error rates for $|\mathcal{X}| = 256$.

to transform the original data space into an *easier* (i.e., sparser, structured) code space, which is then quantized and used as the interface for the communication channel [2]. In fact, this transformation aims to exploit the inherent structure and correlations within the source data, effectively concentrating the original signal into a smaller number of coefficients. This concentration consequently enables more efficient quantization and encoding, leading to significant compression gains. Again, in transform coding is also important to define utility metrics that will be optimized through the transform. For example, the rate-distortion tradeoff is considered in reconstruction problems, where the rate represents the number of bits spent to represent the compressed signal, and the distortion measures the quality of the final reconstruction.

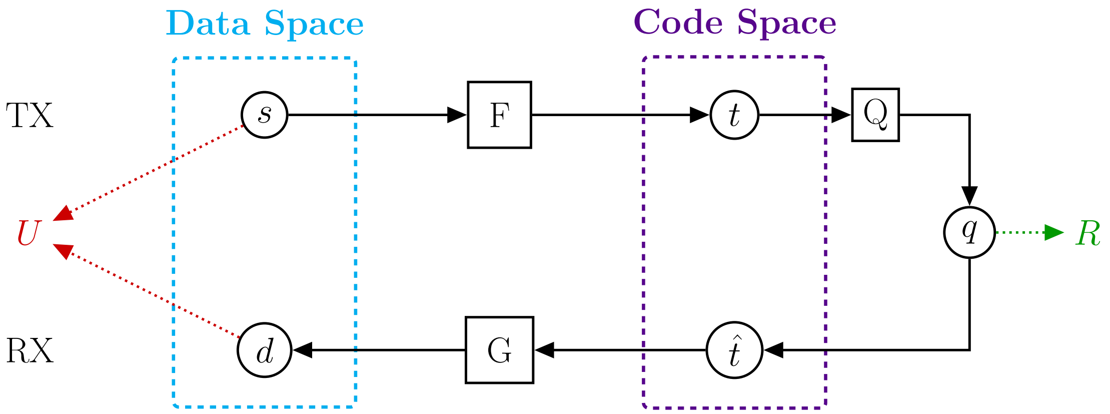


Figure 2.7: Transform coding framework adapted from [2, 3]. Circles denote variables, while squares denote functions. The upper branch represents the transmitter, while the lower branch is the receiver. A source random variable s is mapped into the *code space* through a transformation $t = \mathrm{F}(s)$. The transformed variable t is then quantized into a discrete variable $q = \mathrm{Q}(t)$, which is compressed at a rate R and sent through the communication channel. The receiver reconstructs $\hat{t}$ and transforms it back to the data space $d = \mathrm{G}(\hat{t})$. The source and destination variables (s, d) are used to compute the utility function U. The end-to-end optimization minimizes $L = R - \lambda U$.

A pictorial block diagram for transform coding is shown in Fig. 2.7, where the upper branch represents the source (or transmitter), while the lower branch represents the destination (or receiver). A source random variable s is mapped into the *code space* through a transformation $t = \mathrm{F}(s)$. The transformed variable t is then quantized into a discrete variable $q = \mathrm{Q}(t)$, through the quantization function Q. Note that in general, the quantization is a lossy operation. The discrete variable q is then compressed at a rate R and sent through the communication channel, where R is the number of bits required for the data transmission. For example, entropy coding allows to compress q in a lossless fashion at a rate close to its entropy $R \approx H(P_q)$. The receiver reconstructs $\hat{t}$, and transforms it back to the data space $d = \mathrm{G}(\hat{t})$ through the *inverse* transform G. The source and destination variables (s, d) are used to compute the utility function U. The goal of the system designer is to minimize the rate-utility tradeoff

$$L = R - \lambda U, \tag{2.21}$$

where the coefficient λ is used to weigh the importance of the utility in the optimization.

In the next example (adapted from [26]), we provide a simple connection between the abovementioned transform coding setup (Fig. 2.7) and the channel state information (CSI) problem. A more comprehensive CSI use case is also presented in Chapter 3.

Example 1. *Consider the CSI feedback problem: s and d are channel values in the frequency-space (antenna) domain; the transforms* F *and* G *are Fourier and inverse Fourier transform, respectively. Hence, the code space represents channel value in the angular-delay domain – which is a sparser representation in most use cases. The discrete variable q is a quantized version of t: the rate R depends on the probability mass function of the variable q. When interested in channel reconstruction (i.e., $d = \hat{s}$), the (negative) mean-squared error (MSE) can be used as the utility function. The optimization tradeoff in this case is $L = R + \lambda \cdot MSE(s, d)$.*

Note that standard transform coding implies strict modularity, which means that the processing blocks (transform, quantization, entropy coding) operate independently. Linear transformations have been a popular choice to convert the original data into the code space since they usually provide simple implementations. Fourier and discrete cosine transforms have been at the basis for several image, audio, and video compression standards since they are designed to concentrate the energy of the signal into low-frequency coefficients. This property is crucial for efficient compression, as it allows for coarsely quantizing the high-frequency coefficients with minimal impact on the quality. For example, the JPEG [27] image compression algorithm relies on discrete-cosine transform followed by quantization. Then, lossless entropy coding is used to efficiently transmit/store the quantized representations.

2.3 Neural Compression

Recent advantages in signal processing showed that nonlinear transform coding (NTC) outperforms linear transformation in a wide range of applications [28]. In the NTC case, the transformation functions (F, G) are nonlinear functions. Since it is known that neural networks provide "universal" function approximators [29], recent NTC-based compressors model the transformations as neural networks $(\mathrm{F}_\theta, \mathrm{G}_\phi)$, where θ and ϕ represent the neural network parameters. When trained with real-world data in an end-to-end fashion, neural network allow for great expressivity and flexibility in the transform function definitions.

When focusing on the end-to-end optimization of communication systems with machine learning methods, a differentiable formulation must be provided for all the building blocks of the transmitter-receiver chain. When the goal is to minimize the rate-utility tradeoff $L = R - \lambda U$, one has to provide differentiable methods to compute the gradients of L with respect to the model parameters. The optimization of the utility function U is a well-established concept in machine learning literature. The main idea is to identify a performance metric and train the system to optimize such a metric (or a proxy for it). For instance, MSE is commonly used for both training and testing in reconstruction (regression) problems. However, in

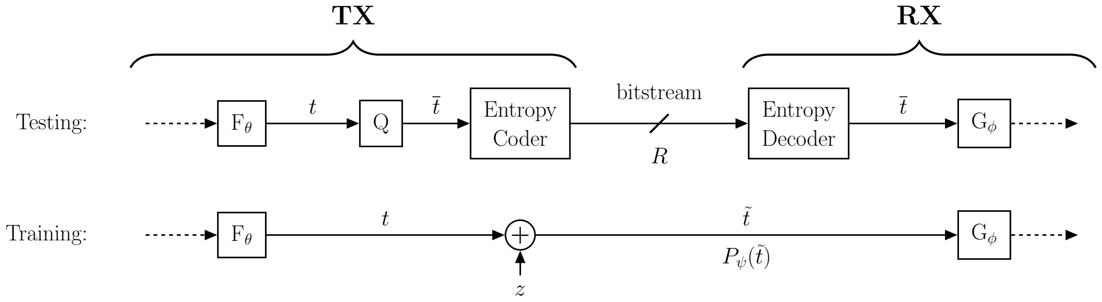


Figure 2.8: Rate optimization block diagram, during testing (top) and training (bottom).

classification problems, the non-differentiable nature of the error rate leads to the use of cross-entropy as a proxy for maximizing accuracy.

On the other hand, the optimization of the communication overhead R was introduced in [3, 30] for a popular class of neural compressors for images. Following the diagram in Fig. 2.8, we here summarize the main processing blocks that are used during testing and training.

During testing: the continuous variable t is quantized to the closest integer. The resulting discrete variable is denoted by $\bar{t}$. The entropy coding block converts $\bar{t}$ into a bitstream; the probability distribution of $\bar{t}$, denoted by $P(\bar{t})$, is assumed to be estimated (with high fidelity) from training. Let $H[P(\bar{t})]$ be the entropy of the discrete random variable $\bar{t}$; then, the entropy coder will produce a bitstream with rate $R \approx H[P(\bar{t})]$ bits/channel use.

During training: the quantization and entropy coding blocks cannot be used since they are not differentiable operations. Instead, these blocks are replaced by i.i.d. uniform noise, which simulates quantization noise introduced by rounding to the closest integer [3, 30]. The *pseudo-quantized* variable is $\tilde{t} = t + z$, where $z \sim \mathcal{U}[-0.5, 0.5]$ is the quantization noise. The distribution of $\tilde{t}$, denoted by $P_\psi(\tilde{t})$, is learned during training, where the parameters are indicated by ψ. Note that

$P_\psi(\tilde{t})$ is a continuous relaxation of $P(\bar{t})$ [3].

Finally, given the parameterized distribution $P_\psi(\tilde{t})$, the rate component can be expressed as

$$R = -\mathbb{E}[\log_2 P_\psi(\tilde{t})] \tag{2.22}$$

and it represents the number of bits transmitted for each channel use.

The method outlined above (or variations of it) is one of our building blocks for the learned task-aware compression settings analyzed in the next chapters.

Chapter 3

Precoding-Oriented CSI Feedback

In frequency division duplexing systems, downlink massive multiple-input multiple-output (MIMO) precoding algorithms rely on accurate channel state information (CSI) feedback from users. This thesis investigates the tradeoff between the CSI feedback overhead and the resulting user performance in terms of achievable rate. The goal is to determine the beamforming information (precoding) directly from the user feedback. We employ a deep learning-based approach to design an end-to-end precoding-oriented feedback architecture, including learned pilots, user compressors for finite-rate feedback, and base station processing to determine precoding vectors. We propose a novel loss function that maximizes the sum of achievable rates while minimizing the CSI feedback overhead. We consider both single- and multi-cell multi-user MIMO systems, analyzing the impact of intra- and inter-cell interference on the CSI feedback strategy design. Simulation results demonstrate that our approach outperforms previous precoding-oriented methods and offers greater efficiency than conventional methods that separate CSI

Part of this work was presented at IEEE ICC 2023 [31].

compression and precoding.

3.1 Introduction

Massive multiple-input multiple-output (MIMO) is a fundamental technology of 5G and of future wireless systems, and accurate channel state information (CSI) is a key enabler to unlock its full potential [32, 33]. As we move towards 6G, with larger antenna arrays and wider bandwidths, effective CSI becomes even more critical for achieving the desired performance. When operating in time division duplexing, the base station (BS) leverages channel reciprocity to obtain CSI from uplink transmissions. However, this reciprocity does not hold in frequency division duplexing (FDD), requiring users to estimate the downlink channel realizations and feed back (on the uplink) the estimated downlink CSI to the BS, incurring a communication overhead. This feedback burden becomes increasingly taxing as the system dimensions increase (e.g., more antennas, more users, more subcarriers), impacting the scalability of FDD systems.

The CSI feedback challenges become even more prohibitive when considering multi-cell systems, where the users experience inter-cell interference in addition to the intra-cell one. Multi-cell MIMO cooperation, in its simplest form, enables the BSs to share the CSI for interference coordination [34]. The CSI is shared across BSs through backhaul links allowing BSs to coordinate their signaling strategies (e.g., beamforming, scheduling) for interference avoidance. This basic level of coordination requires a relatively small amount of backhaul communication (only the CSI) since information-carrying signals are not shared. For instance, each BS has to know the CSI for the desired users (served in the same cell), and the interfering users that are served by BSs in neighboring cells. In this case, each user may transmit both the desired and interfering CSI feedback to the serving BS, then the BS will forward (through the backhaul link) the appropriate interfering CSI information to the adjacent BSs [34].

The CSI feedback compression problem has been widely investigated in the past. Traditionally, the primary focus has been on the channel reconstruction problem, aiming to reproduce the user's estimated downlink channel at the base station, which the BS then uses to do multi-user precoding [35, 36]. Classical CSI compression techniques make use of signal processing techniques such as vector quantization [37] and compressed sensing [38]. In the first case, the overhead is still impractical for large systems, while the latter technique assumes channel sparsity in some domains.

Depending on the type of MIMO processing adopted in the system, three fundamental metrics can be considered for the CSI feedback problem: (i) *overhead*, that is the number of bits sent on the feedback link; (ii) *performance*, that is the total achievable rate at the users; (ii) *distortion*, that is the loss (e.g., mean squared error) incurred when trying to reconstruct the channel realizations at the BS. For reference, classical methods have been focused on the channel reconstruction problem, i.e., the focus is on optimizing the overhead-distortion (or rate-distortion) tradeoff. The reconstructed channels are then used to design the downlink signals. In broadcast (downlink) MIMO channels, determining the best precoding method for optimal performance requires complex, non-linear calculations that become increasingly difficult as the network size grows [39, 40, 41]. Generally speaking, it is known that the optimal downlink rate is achieved with nonlinear precoding methods [36, 42] based on dirty paper coding [43]. In this thesis, instead, we assume that transmit beamforming is implemented with linear precoding as it is a popular choice due to its simple implementation [35]. Shifting toward a task-oriented approach, we are interested in directly optimizing the overhead-performance tradeoff when designing the compression algorithm.

Recently, deep learning-based solutions have been proposed for the CSI feedback problem in massive MIMO FDD systems, see [44] and references therein for an overview. Concurrently, the 3rd Generation Partnership Project (3GPP) identified the CSI feedback framework as one of the first use cases for the integration between machine learning and artificial intelligence in wireless networks [45]. In general, these data-driven solutions rely on fewer assumptions and outperform classical CSI feedback methods. The capability of the autoencoder structure [46] for the CSI compression problem was first shown by [26]. Several follow-up works focused on channel reconstruction have improved the distortion metric [44] with increasing dimensionality reduction on the feedback link. Similarly to image processing applications [3, 30], a mechanism for the feedback overhead optimization has been introduced in [47]. The authors [47] consider the rate-distortion objective, where the goal is to reconstruct channel realizations with minimal overhead at the BS side. Their results [47] show that the feedback overhead can be further reduced with respect to previous work, but there is no focus on the final task that is performed at the BS (e.g., beamforming).

On the other hand, [4, 48, 49, 50] consider objectives related to the final task to be performed at the BS, i.e., beamforming with linear precoding. A single-user system is analyzed in [48], where precoding information is computed and compressed at the user side, and decoded at the base station: the proposed solution shows overhead savings compared with 3GPP standards methods. Instead, [4, 49, 50] proposed a beamforming-oriented architecture that includes the downlink channel estimation phase (done with pilots), the uplink feedback compression, and the computation of the downlink precoding. The end-to-end optimization results in learned pilots, learned UE processing to extract beamforming-oriented features from the received pilots, and learned BS processing to determine the downlink beamforming from the compressed CSI. In [4], a single-cell multi-user narrowband system is considered, where the system output is the collection of linear precoders and the objective function is the sum of achievable rates experienced by the users. Their best results [4, Fig. 9] are obtained by modeling each feedback tap as a binary value, using a smooth approximation during training to allow backpropagation. Hence, the feedback overhead is determined by the choice of the feedback dimension (fixed), without the possibility to obtain further compression. In [50] instead, a single-user MIMO with orthogonal frequency division multiplexing (OFDM) modulation and hybrid analog-digital beamforming architecture is considered. Their feedback compression relies on the vector quantized variational autoencoder (VQ-VAE) [51], where the loss function aims to construct an optimized codebook of fixed size. In our previous work [49], we focused on a single-cell multi-user system, while in this thesis we extend the framework to a multi-cell multi-user system.

Extending the CSI feedback problem to multi-cell systems, previous work [52] considered the optimization of the *soft-handoff* model, where a single user per cell gets interference from one neighboring cell only. In that case, there is no intra-cell interference, but only inter-cell interference. Moreover, the authors in [52] do not consider the channel estimation phase and do not explicitly optimize the CSI feedback overhead, since they consider a pre-defined feedback size with post-training uniform quantization, without specifically taking the overhead into account in their loss function. Another problem that arises in multi-cell systems is how to split the CSI overhead between the desired and the interfering channel. A feedback-bit allocation strategy for the quantized CSI feedback under the soft-handoff model was proposed in [53], leveraging random vector quantization (RVQ) for CSI compression. The authors [53] focus on maximizing sum-rate in Rayleigh fading channels using quantized CSI, employing a high SINR approximation.

In this thesis, we analyze the tradeoff between the CSI feedback overhead and the resulting sum of achievable user rates in a system employing linear precoding on the downlink. Our task-oriented approach focuses on the BS determining precoding vectors for each user to maximize the sum of achievable rates. We extend our previous work [49] to include multi-cell multi-user systems, which are affected by both intra-cell and intra-cell interference. We assume that the BSs in this multi-cell system cooperate for interference coordination [34], by sharing CSI information that helps design proper precoding vectors for the users. Our proposed loss function includes the optimization of the overhead required for the CSI (number of feedback bits), by leveraging recent neural compression methods from the image processing literature [3, 30]. In our end-to-end architecture, similar to [4], the users observe a sequence of noisy pilots as input, and produce precoding-oriented feedback messages for the BS. The BS processes the received feedback and determines the precoder vectors for each user. The pilots, feedback schemes, and BS processing are modeled with neural networks. Differently from [4], we include a feedback overhead optimization mechanism that estimates the feedback distribution during training and uses entropy coding to generate the bit streams at test time [3, 30]. The entropy of the feedback values is used to estimate the feedback overhead (feedback rate). For the multi-cell scenario, we do not impose a structure on the splitting between desired and interfering channels, but we let the optimization determine the final CSI overhead allocations. In order to train the end-to-end architecture with gradient descent, we propose a tunable loss function that captures the tradeoff between feedback overhead and system performance. Note that by designing an appropriate loss function to optimize the overhead-performance tradeoff, this approach can be applied to any neural network architecture, the choice of which is typically dictated by the nature of the input data. We show that the precoding-oriented system trained using the overhead-performance loss function outperforms conventional methods based on channel reconstruction followed by traditional precoding techniques. In our precoding-oriented approach, the user network is able to learn how to efficiently transfer precoding-oriented quantized information over the feedback link. We also address several robustness questions for practical deployments: (i) we analyze the scalability of our system with the number of users and cells, and propose learning strategies to facilitate the training orchestration of dense networks; (ii) we investigate the robustness of the learned solution with respect to the deployment scenario, focusing on asymmetric scenarios where users experience different channel conditions.

We illustrate the system model in Section 3.2, while the precoding-oriented CSI feedback approach is discussed in Section 3.3. Numerical results are shown in Section 3.4, while conclusions are drawn in Section 3.5.

3.2 System Model

We consider a multi-cell multi-user massive MIMO system where multiple BSs serve multiple users geographically distributed over different cells. Although our

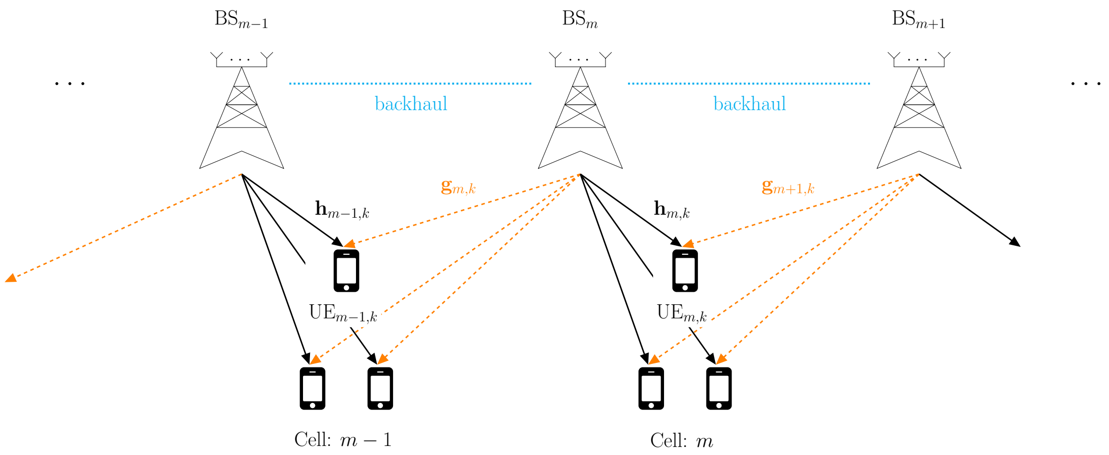


Figure 3.1: Representation of the downlink signals for the multi-cell multi-user system for the soft-handoff model, where users are assumed to be at the cell edge. This illustration focuses on the $(m-1)$-th and the m-th cell. The black solid arrows denote the desired channels $\mathbf{h}$, while the orange dashed arrows denote the interfering channels $\mathbf{g}$.

approach applies to general multi-cell multi-user systems, for notation convenience we present a modified soft-handoff model [40], a variant of the linear Wyner model [54]. The system consists of M cells arranged circularly, with each cell indexed by $m \in \{1, \ldots, M\}$ and containing one BS equipped with N_t transmit antennas. Each cell serves K single-antenna users, for a total of $M \cdot K$ users across the multi-cell system. We assume that users within a cell experience inter-cell interference only from the BS located immediately *to their right*, as illustrated in Figure 3.1 for two adjacent cells. This is a typical assumption for the soft-handoff model [40], since handoff situations are a common occurrence in real-life cellular systems. In other words, the users in the m-th cell suffer interference from the $(m+1)$-th BS; the users in the M-th cell suffer interference from the 1-st BS.

In our system, we adopt multi-cell cooperation specifically for *interference coordination*, as detailed in [34]. This approach requires the BSs to share downlink CSI for both desired and interfering links. This CSI is acquired through user feedback on the uplink channel. Note that this level of coordination necessitates backhaul links between the BSs to allow for the CSI exchange. It is important to remark that, in this interference coordination mode, only CSI is shared between BSs – not the actual user data.

We assume that linear precoding is used at each BS, hence the downlink transmitted signal for the m-th BS is

$$\mathbf{x}_m = \sum_{k=1}^{K} \mathbf{v}_{m,k} s_{m,k} = \mathbf{V}_m \mathbf{s}_m \tag{3.1}$$

where $\mathbf{v}_{m,k} \in \mathbb{C}^{N_t}$ is the precoding vector and $s_{m,k} \in \mathbb{C}$ is the symbol to be sent for the k-th user in the m-th cell. Each precoding matrix $\mathbf{V}_m \in \mathbb{C}^{N_t \times K}$ satisfies the power constraint $\mathrm{Tr}(\mathbf{V}_m \mathbf{V}_m^H) \leq P$, and the symbols $\mathbf{s}_m \in \mathbb{C}^K$ are normalized to $\mathbb{E}[\mathbf{s}_m \mathbf{s}_m^H] = \mathbf{I}$, for $m = 1, \ldots, M$.

We assume that the desired and interfering signal powers received by the k-th user in the m-th cell are determined by the user's location within the cell. The transmitted signals experience both small-scale and large-scale fading, including distance-dependent path loss and shadowing effects. After averaging over small-scale fading, the resulting desired and interfering signal powers are denoted by $\gamma_{m,k}$ and $\eta_{m,k}$, respectively. Hence, the received signal at the k-th user in the m-th cell is

$$\begin{aligned} y_{m,k} = {} & \sqrt{\gamma_{m,k}}\, \mathbf{h}_{m,k}^H \mathbf{v}_{m,k} s_{m,k} + \sqrt{\gamma_{m,k}} \sum_{j \neq k} \mathbf{h}_{m,k}^H \mathbf{v}_{m,j} s_{m,j} \\ & + \sqrt{\eta_{m,k}} \sum_{i} \mathbf{g}_{m+1,k}^H \mathbf{v}_{m+1,i} s_{m+1,i} + z_{m,k} \end{aligned} \tag{3.2}$$

where $\mathbf{h}_{m,k} \in \mathbb{C}^{N_t}$ is the vector of downlink channel gains from the desired BS, namely BS m, for the k-th user in the m-th cell; $\mathbf{g}_{m+1,k} \in \mathbb{C}^{N_t}$ is the interfering channel gain from the adjacent cell for the k-th user in the m-th cell; $z_{m,k} \sim \mathcal{CN}(0, \sigma^2_{m,k})$ is the additive white Gaussian noise. Let $\eta_{m,k} = \alpha_{m,k}\ \gamma_{m,k}$, with $\alpha_{m,k} \in [0, 1]$, and $\rho_{m,k} = \gamma_{m,k}/\sigma^2_{m,k}$. In other words, $\alpha_{m,k}$ represents the ratio between the power of the desired signal and the interfering signal from adjacent cells, for the k-th user in the m-th cell. On the other hand, $\rho_{m,k}$ represents the signal-to-noise ratio (SNR) of the received signal and it is independent of beamforming vectors. We assume that $\rho_{m,k}$ is known at the BS; in 3GPP context, it is analogous to the *channel quality indicator* (CQI) [55]. Note that a single interfering signal is reported in (3.2) since the soft-handoff model is considered. In the case of the general multi-cell multi-user system, multiple interfering terms scaled by channel gains $\mathbf{g}_{t,k}$, $t \neq m + 1$, may be added to (3.2).

Given the received signal model in (3.2), the signal-to-interference-and-noise-ratio (SINR) for the k-th user in the m-th cell is defined as

$$\mathrm{SINR}_{m,k} = \frac{|\mathbf{h}^H_{m,k}\mathbf{v}_{m,k}|^2}{\mathrm{INTRA}_{m,k} + \mathrm{INTER}_{m,k} + 1/\rho^2_{m,k}},$$

where

$$\begin{aligned} \mathrm{INTRA}_{m,k} &= \sum_{j \neq k} |\mathbf{h}^H_{m,k}\mathbf{v}_{m,j}|^2, \\ \mathrm{INTER}_{m,k} &= \alpha_{m,k} \sum_{i} |\mathbf{g}^H_{m+1,k}\mathbf{v}_{m+1,i}|^2. \end{aligned} \tag{3.3}$$

Note that $\mathrm{INTRA}_{m,k}$ represents the intra-cell interference, while $\mathrm{INTER}_{m,k}$ quantifies the inter-cell interference, experienced by the k-th user in the m-th cell.

The achievable rate for the k-th user in the m-th cell is

$$R_{m,k} = \log_2\left(1 + \mathrm{SINR}_{m,k}\right). \tag{3.4}$$

Considering the multi-cell multi-user communication problem with M cells, each with K users, the system performance is characterized by the network sum rate, namely the sum of all the achievable rates, i.e.,

$$R = \sum_{m=1}^{M} \sum_{k=1}^{K} R_{m,k}. \tag{3.5}$$

The precoding matrices $\mathbf{V}_m$, $m = 1, \ldots, M$, are jointly optimized at the network level, thus accurate CSI is essential for designing the $\mathbf{V}_m$'s that maximize the network sum rate in (3.5). Ideally, each BS should beamform its signals to the desired users while simultaneously minimizing the interference towards users in neighboring cells. We assume that neither the BSs nor the users have prior knowledge of the channel realizations. Therefore, each of the $m = 1, \ldots, M$ BSs must determine its corresponding downlink precoding matrix $\mathbf{V}_m$ based only on the feedback received from the users on the uplink. Note that knowledge of both $\mathbf{h}$ and $\mathbf{g}$ is needed to maximize the SINR in (3.3). Consequently, we assume that each BS sends reference signals (pilots) during the downlink channel estimation phase. We assume that orthogonal scheduling is adopted to coordinate the downlink channel estimation phase, e.g., time scheduling, so that each user is able to distinguish between signals from different base stations. The pilots for the m-th BS, of length L, are denoted by $\tilde{\mathbf{X}}_m \in \mathbb{C}^{N_t \times L}$. The received noisy pilots at the k-th user in m-th cell are

$$\tilde{\mathbf{y}}^{\mathrm{D}}_{m,k} = \sqrt{\gamma_{m,k}}\, \mathbf{h}^H_{m,k} \tilde{\mathbf{X}}_m + \mathbf{z}^{\mathrm{D}}_{m,k}, \tag{3.6}$$

$$\tilde{\mathbf{y}}^{\mathrm{I}}_{m,k} = \sqrt{\eta_{m,k}}\, \mathbf{g}^H_{m+1,k} \tilde{\mathbf{X}}_{m+1} + \mathbf{z}^{\mathrm{I}}_{m,k}, \tag{3.7}$$

where $\mathbf{z}_{m,k} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ is the additive white Gaussian noise at the k-th user in the m-th cell; $\tilde{\mathbf{y}}^{\mathrm{D}}_{m,k}$ denotes the received signal when pilots from the serving

BS are transmitted, while $\tilde{\mathbf{y}}^{\mathrm{I}}_{m,k}$ denote the received signal when pilots from the interfering (adjacent) BS are transmitted. The ℓ-th pilot transmission satisfies the instantaneous power constraint $\|\tilde{\mathbf{x}}_{m,\ell}\|_2^2 \leq P$, where $\tilde{\mathbf{x}}_{m,\ell}$ is the ℓ-th column of $\tilde{\mathbf{X}}_{\mathbf{m}}$.

In general, the CSI feedback overhead is proportional to the dimensions of the system (e.g., number of users, antennas), and becomes very large in case of massive MIMO systems with many users. Hence, users are required to extract relevant CSI from the received pilots, then feed back the compressed CSI, or other relevant information needed for the downlink precoding, to the BS over a rate-limited link. The feedback for the user k-th in the m-th cell is as follows:

$$\mathbf{b}^{\mathrm{D}}_{m,k} = \mathcal{F}^{\mathrm{D}}_{m,k}(\tilde{\mathbf{y}}^{\mathrm{D}}_{m,k}), \tag{3.8}$$

$$\mathbf{b}^{\mathrm{I}}_{m,k} = \mathcal{F}^{\mathrm{I}}_{m,k}(\tilde{\mathbf{y}}^{\mathrm{I}}_{m,k}), \tag{3.9}$$

where $\mathcal{F}^{\mathrm{D}}_{m,k}$ and $\mathcal{F}^{\mathrm{I}}_{m,k}$ are the feedback schemes (compressors) used for the desired and interfering CSI, respectively; $\mathbf{b}^{\mathrm{D}}_{m,k}$ is the feedback for the desired BS, and $\mathbf{b}^{\mathrm{I}}_{m,k}$ is the feedback for the interfering BS.

Following the multi-cell MIMO cooperation for interference mitigation approach, we assume that each BS forwards the interfering CSI to the adjacent BSs through backhaul links, as depicted in Fig. 3.2.

Therefore, the m-th BS collects the desired feedback from the K users in the m-th cell, denoted by $(\mathbf{b}^{\mathrm{D}}_{m,1}, \ldots, \mathbf{b}^{\mathrm{D}}_{m,K})$, and receives the interfering feedback $(\mathbf{b}^{\mathrm{I}}_{m-1,1}, \ldots, \mathbf{b}^{\mathrm{I}}_{m-1,K})$ from the K users in the $(m-1)$-th cell through the backhaul.

Hence, each of the $m = 1, \ldots, M$ BS computes the precoding matrices as

$$\mathbf{V}_m = \mathcal{G}_m \left[(\mathbf{b}^{\mathrm{D}}_{m,1}, \ldots, \mathbf{b}^{\mathrm{D}}_{m,K}), (\mathbf{b}^{\mathrm{I}}_{m-1,1}, \ldots, \mathbf{b}^{\mathrm{I}}_{m-1,K}) \right], \tag{3.10}$$

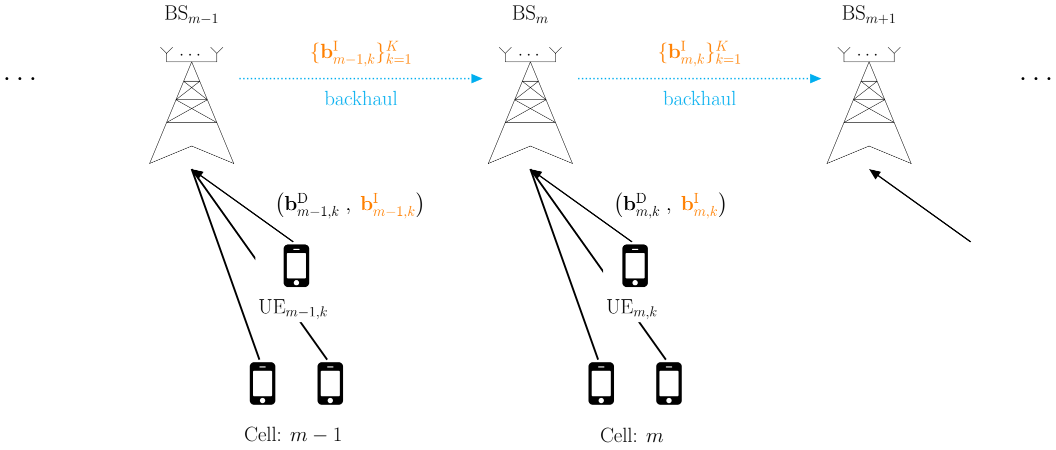

BS_{m-1}
BS_m
BS_{m+1}
$\{\mathbf{b}^{\mathrm{I}}_{m-1,k}\}_{k=1}^{K}$
backhaul
$\{\mathbf{b}^{\mathrm{I}}_{m,k}\}_{k=1}^{K}$
backhaul
$(\mathbf{b}^{\mathrm{D}}_{m-1,k}\ ,\ \mathbf{b}^{\mathrm{I}}_{m-1,k})$
$(\mathbf{b}^{\mathrm{D}}_{m,k}\ ,\ \mathbf{b}^{\mathrm{I}}_{m,k})$
$\mathrm{UE}_{m-1,k}$
$\mathrm{UE}_{m,k}$
Cell: $m-1$
Cell: m

Figure 3.2: Representation of the uplink feedback for the multi-cell multi-user soft-handoff model, focusing on the $(m-1)$-th and the m-th cell. Both desired CSI $\mathbf{b}^{\mathrm{D}}$ and the interfering CSI $\mathbf{b}^{\mathrm{I}}$ are sent on the uplink to the serving cell. The interfering CSI $\mathbf{b}^{\mathrm{I}}$ (in orange) is forwarded to the adjacent cell through the backhaul link (in blue).

where $\mathcal{G}_m$ denotes the BS processing. Note that in our setup only $\mathbf{V}_m$ is provided at the output of each BS, as the ultimate task of maximizing sum rate in (3.5) only depends on $\mathbf{V}_m$. Potentially, each BS could also provide channel reconstructions for $\mathbf{h}$ and $\mathbf{g}$ that could be used for traditional precoding techniques.

Our focus in this thesis is to design a precoding-oriented CSI feedback strategy, where the tradeoff between the feedback overhead and the network sum rate is considered. Our system seeks to maximize the sum of achievable rates (3.5), while the amount of bits required to transmit $(\mathbf{b}^{\mathrm{D}}_{m,1}, \ldots, \mathbf{b}^{\mathrm{D}}_{m,K})$ and $(\mathbf{b}^{\mathrm{I}}_{m,1}, \ldots, \mathbf{b}^{\mathrm{I}}_{m,K})$ over the feedback link is bounded. We use neural networks to design the pilots $\tilde{\mathbf{X}}_m$, the feedback schemes $\mathcal{F}^{\mathrm{D}}$ and $\mathcal{F}^{\mathrm{I}}$, and the BS processing $\mathcal{G}$.

3.3 Precoding-oriented CSI Feedback with Overhead-performance Tradeoff

As explained in the previous section, we define three stages for the precoding-oriented CSI feedback strategy.

1. Downlink channel estimation phase, where the desired and interfering pilots are received at each user according to (3.6) and (3.7), respectively.

2. Uplink CSI feedback phase, where the desired and interfering CSI information is computed by each user and sent according to (3.8) and (3.9), respectively.

3. Downlink precoding phase, where each BS computes the precoding vectors according to (3.10).

We use neural networks in place of conventional methods for the pilots $\{\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_M\}$, the feedback schemes $\{\mathcal{F}^{\mathrm{D}}, \mathcal{F}^{\mathrm{I}}\}$, and the BS schemes $\mathcal{G}$. We also adopt a mechanism, proposed in [3, 30], that optimizes the compression and quantization of the feedback. We propose a precoding-oriented loss function, where the overhead-performance tradeoff is directly embedded in the objective function. In this way, the feedback scheme $\{\mathcal{F}^{\mathrm{D}}_{m,k}, \mathcal{F}^{\mathrm{I}}_{m,k}\}$ can effectively extract an efficient precoding-oriented representation of the channel realizations, and the BS processing $\mathcal{G}$ is able to directly output the precoding matrices $\mathbf{V}_m$'s. When it is clear from the context, we may omit the superscripts $^{\mathrm{D}}$ and $^{\mathrm{I}}$ that indicate the desired and interfering CSI processing blocks, respectively. More details about each of the processing blocks are provided in the next sections.

An illustration of the block diagram is reported in Fig. 3.3. The blocks in green denote the UE side, while the blocks in red denote the BS side. More details about

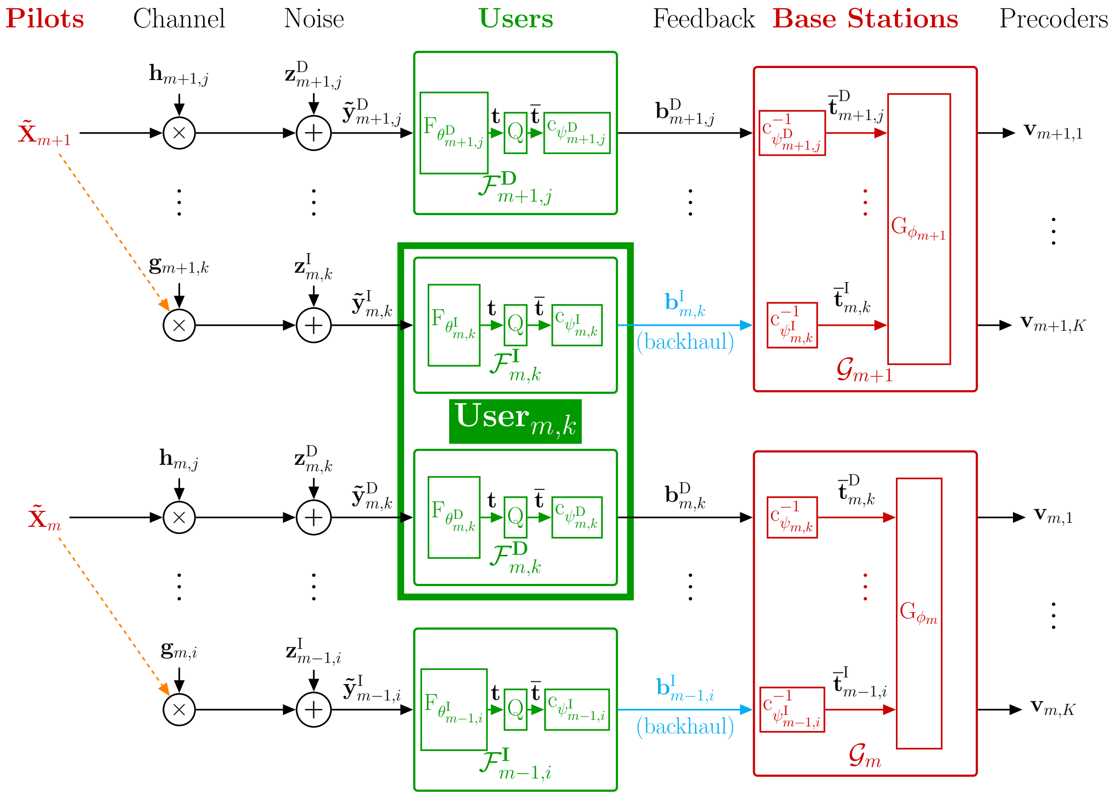

Pilots
Channel
Noise
Users
Feedback
Base Stations
Precoders
$\tilde{\mathbf{X}}_{m+1}$
$\mathbf{h}_{m+1,j}$
$\mathbf{z}^{\mathrm{D}}_{m+1,j}$
$\tilde{\mathbf{y}}^{\mathrm{D}}_{m+1,j}$
$\mathrm{F}_{\theta^{\mathrm{D}}_{m+1,j}}$
$\mathbf{t}$
Q
$\bar{\mathbf{t}}$
$\mathrm{c}_{\psi^{\mathrm{D}}_{m+1,j}}$
$\mathcal{F}^{\mathbf{D}}_{m+1,j}$
$\mathbf{b}^{\mathrm{D}}_{m+1,j}$
$\mathrm{c}^{-1}_{\psi^{\mathrm{D}}_{m+1,j}}$
$\bar{\mathbf{t}}^{\mathrm{D}}_{m+1,j}$
$\mathrm{G}_{\phi_{m+1}}$
$\mathbf{v}_{m+1,1}$
$\mathbf{g}_{m+1,k}$
$\mathbf{z}^{\mathrm{I}}_{m,k}$
$\tilde{\mathbf{y}}^{\mathrm{I}}_{m,k}$
$\mathrm{F}_{\theta^{\mathrm{I}}_{m,k}}$
$\mathrm{c}_{\psi^{\mathrm{I}}_{m,k}}$
$\mathcal{F}^{\mathbf{I}}_{m,k}$
$\mathbf{b}^{\mathrm{I}}_{m,k}$
(backhaul)
$\mathrm{c}^{-1}_{\psi^{\mathrm{I}}_{m,k}}$
$\bar{\mathbf{t}}^{\mathrm{I}}_{m,k}$
$\mathbf{v}_{m+1,K}$
$\mathcal{G}_{m+1}$
$\mathrm{User}_{m,k}$
$\tilde{\mathbf{X}}_{m}$
$\mathbf{h}_{m,j}$
$\mathbf{z}^{\mathrm{D}}_{m,k}$
$\tilde{\mathbf{y}}^{\mathrm{D}}_{m,k}$
$\mathrm{F}_{\theta^{\mathrm{D}}_{m,k}}$
$\mathrm{c}_{\psi^{\mathrm{D}}_{m,k}}$
$\mathcal{F}^{\mathbf{D}}_{m,k}$
$\mathbf{b}^{\mathrm{D}}_{m,k}$
$\mathrm{c}^{-1}_{\psi_{m,k}}$
$\bar{\mathbf{t}}^{\mathrm{D}}_{m,k}$
$\mathbf{v}_{m,1}$
$\mathrm{G}_{\phi_{m}}$
$\mathbf{g}_{m,i}$
$\mathbf{z}^{\mathrm{I}}_{m-1,i}$
$\tilde{\mathbf{y}}^{\mathrm{I}}_{m-1,i}$
$\mathrm{F}_{\theta^{\mathrm{I}}_{m-1,i}}$
$\mathrm{c}_{\psi^{\mathrm{I}}_{m-1,i}}$
$\mathcal{F}^{\mathbf{I}}_{m-1,i}$
$\mathbf{b}^{\mathrm{I}}_{m-1,i}$
(backhaul)
$\mathrm{c}^{-1}_{\psi^{\mathrm{I}}_{m-1,i}}$
$\bar{\mathbf{t}}^{\mathrm{I}}_{m-1,i}$
$\mathbf{v}_{m,K}$
$\mathcal{G}_{m}$

Figure 3.3: Illustration of the learned precoding-oriented CSI feedback system model, focusing on the k-th user in the m-th cell. The pilots (marked in red) are learned during the end-to-end learning. The green boxes correspond to the users' feedback schemes for the desired ($\mathcal{F}^{\mathrm{D}}$) and interfering ($\mathcal{F}^{\mathrm{I}}$) CSI. Each feedback scheme comprises the neural network F_θ, the quantizer Q, and the entropy coder c_ψ. The BS processing is denoted with $\mathcal{G}$, which is composed by the entropy decoder c^{-1}_ψ and the DNN G_ϕ.

each processing block are provided in the remainder of this section.

3.3.1 Downlink Pilots

The downlink received pilots for the k-th user in the m-th cell are expressed in (3.6) and (3.7). As in [4], we model each pilot $\tilde{\mathbf{X}}_m$ as the output of a fully connected neural network (single layer) with linear activation and zero bias. The power constraint P is guaranteed during training by setting $\|\tilde{\mathbf{x}}_{m,\ell}\|_2^2 = P$, $\ell =$

$1, \ldots, L$, where $\tilde{\mathbf{x}}_{m,\ell}$ represents the ℓ-th column of $\tilde{\mathbf{X}}_m$, and L is the pilot duration. Gaussian noise is added to the sequence of L received pilots at the users to model the receiver noise according to (3.6) and (3.7).

3.3.2 Feedback Scheme

As depicted in Fig. 3.3, the feedback schemes for the desired and interfering CSI for each user are denoted by $\mathcal{F}^{\mathrm{D}}_{m,k}$ and $\mathcal{F}^{\mathrm{I}}_{m,k}$, respectively. Generally speaking, $\mathcal{F}^{\mathrm{D}}_{m,k}$ and $\mathcal{F}^{\mathrm{I}}_{m,k}$ can be different schemes for all the $M \cdot K$ users, since each terminal observes different channel realizations $\mathbf{h}_{m,k}$, $\mathbf{g}_{m,k}$. Both feedback schemes $\mathcal{F}^{\mathrm{D}}_{m,k}$ and $\mathcal{F}^{\mathrm{I}}_{m,k}$ are composed of three components: (i) a neural network $\mathrm{F}_{\theta_{m,k}}$, where $\theta_{m,k}$ is either $\theta^{\mathrm{D}}_{m,k}$ or $\theta^{\mathrm{I}}_{m,k}$; (ii) a quantizer Q; (iii) an entropy coder $\mathrm{c}_{\psi_{m,k}}$, where $\psi_{m,k}$ is either $\psi^{\mathrm{D}}_{m,k}$ or $\psi^{\mathrm{I}}_{m,k}$. More details about (ii)-(iii) are also provided in the next sections.

The neural network $\mathrm{F}_{\theta_{m,k}}$, where $\theta_{m,k}$ represents the set of trainable parameters, is used to extract features from the received pilots $\tilde{\mathbf{y}}_{m,k}$. More precisely, the neural network output for the desired and interfering CSI is

$$\mathbf{t}^{\mathrm{D}}_{m,k} = \mathrm{F}_{\theta^{\mathrm{D}}_{m,k}}(\tilde{\mathbf{y}}^{\mathrm{D}}_{m,k}), \tag{3.11}$$

$$\mathbf{t}^{\mathrm{I}}_{m,k} = \mathrm{F}_{\theta^{\mathrm{I}}_{m,k}}(\tilde{\mathbf{y}}^{\mathrm{I}}_{m,k}), \tag{3.12}$$

where $\mathbf{t}_{m,k} \in \mathbb{R}^{N_b}$, and N_b is the dimension of the neural netowrk output. More details about the neural network architecture are provided in Section 3.4.

The quantizer Q performs uniform scalar quantization to the closest integer. The quantized vector for the k-th user in the m-th cell is denoted as $\bar{\mathbf{t}}_{m,k} = \mathrm{Q}(\mathbf{t}_{m,k})$. During training, the quantization is replaced by adding independent identically distributed (iid) uniform noise $\mathbf{u}_{m,k}$, where the width of the uniform distribution is equal to the quantization bin width, i.e., $u^{1}_{m,k}, \ldots, u^{N_b}_{m,k} \sim \mathcal{U}[-0.5, +0.5]$ [3]. We denote the *pseudo-quantized* vector as $\tilde{\mathbf{t}}_{m,k} = \mathbf{t}_{m,k} + \mathbf{u}_{m,k}$ and it substitutes $\bar{\mathbf{t}}_{m,k}$ during training to allow gradient backpropagation.

Specifically, the pseudo-quantized features for the desired and interfering CSI are

$$\tilde{\mathbf{t}}^{\mathrm{D}}_{m,k} = \mathbf{t}^{\mathrm{D}}_{m,k} + \mathbf{u}^{\mathrm{D}}_{m,k}, \tag{3.13}$$

$$\tilde{\mathbf{t}}^{\mathrm{I}}_{m,k} = \mathbf{t}^{\mathrm{I}}_{m,k} + \mathbf{u}^{\mathrm{I}}_{m,k}. \tag{3.14}$$

The quantized features, instead, are

$$\bar{\mathbf{t}}^{\mathrm{D}}_{m,k} = \mathrm{Q}(\tilde{\mathbf{t}}^{\mathrm{D}}_{m,k}), \tag{3.15}$$

$$\bar{\mathbf{t}}^{\mathrm{I}}_{m,k} = \mathrm{Q}(\tilde{\mathbf{t}}^{\mathrm{I}}_{m,k}). \tag{3.16}$$

The entropy coder $\mathrm{c}_{\psi_{m,k}}$ converts the quantized vector $\bar{\mathbf{t}}_{m,k}$ into bit streams in a lossless fashion, thanks to the set of trainable parameters $\psi_{m,k}$ that models the distribution of $\bar{\mathbf{t}}$ [3]. The parameters for the desired and interfering CSI feedback are denoted by $\psi^{\mathrm{D}}_{m,k}$ and $\psi^{\mathrm{I}}_{m,k}$, respectively. As in [30], $\tilde{\mathbf{t}}_{m,k}$ is modeled using a parametric, fully factorized density function. Each element of $\tilde{\mathbf{t}}_{m,k}$ is modeled as a zero-mean Gaussian distribution with standard deviation learned during training. These learned parameters are then used by c_{ψ} to encode $\bar{\mathbf{t}}_{m,k}$ at test time. Finally, the bitstreams for the desired and interfering CSI are denoted as

$$\mathbf{b}^{\mathrm{D}}_{m,k} = \mathrm{c}_{\psi^{\mathrm{D}}_{m,k}}(\bar{\mathbf{t}}^{\mathrm{D}}_{m,k}), \tag{3.17}$$

$$\mathbf{b}^{\mathrm{I}}_{m,k} = \mathrm{c}_{\psi^{\mathrm{I}}_{m,k}}(\bar{\mathbf{t}}^{\mathrm{I}}_{m,k}). \tag{3.18}$$

3.3.3 Feedback Overhead Optimization

Similarly to previous works in the CSI feedback domain [47], we consider the feedback rate as part of our optimization objective. Since both the quantizer and the entropy coder are not differentiable functions, they are substituted by iid uniform noise during training [3], as described above. The i.i.d. noise $\mathbf{u}_{m,k}$ simulates the quantization noise. The compression performed by the entropy coder is lossless and at a rate close to the entropy of $\bar{\mathbf{t}}_{m,k}$; so, at the BS side we have $\mathrm{c}^{-1}_{\psi_{m,k}}(\mathbf{b}_{m,k}) = \bar{\mathbf{t}}_{m,k}$. In fact, during training, the entropy of $\tilde{\mathbf{t}}_{m,k}$ is estimated in terms of the model parameters $\psi_{m,k}$. Note that the probability density of $\tilde{\mathbf{t}}_{m,k}$ is a continuous relaxation of the probability mass function of $\bar{\mathbf{t}}_{m,k}$ [3], hence the differential entropy of $\tilde{\mathbf{t}}_{m,k}$ approximates the entropy of $\bar{\mathbf{t}}_{m,k}$; the estimated entropy represents the average bit rate at the quantizer output and will be used in our loss function to measure the feedback overhead [3]. During testing, the noise $\mathbf{u}_{m,k}$ is not injected, but the output of F_θ goes through the quantizer Q and entropy coder c_ψ.

Note that our approach is different from [4] and [52]. In [4], the DNN output $\mathbf{t}_{m,k}$ contains binary values, and the dimension of $\mathbf{t}_{m,k}$ determines the feedback overhead. In [52], the output of the encoder network contains real values between -1 and 1 (after *tanh* activation), and it is quantized with uniform quantization of an arbitrary number of bits. However, we argue that further compression of this feedback is possible and can provide significant gains. The authors in [4] also propose an alternative method where $\mathbf{t}_{m,k}$ contains real values that are quantized (using Lloyd's algorithm [56]) according to a given overhead budget, and only the BS DNN is further fine-tuned on the quantized inputs. On the other hand, in our work, the optimization method [30] described above seeks to minimize the feedback entropy (rate) without explicit dependency on the feedback dimensionality. Moreover, our approach allows for end-to-end joint training between pilots, users, and BS processing, including the feedback overhead optimization.

3.3.4 BS Processing

During test time, the entropy decoder at the BS losslessly reconstructs both the received desired and interfering feedback

$$\bar{\mathbf{t}}^{\mathrm{D}}_{m,k} = \mathrm{c}^{-1}_{\psi^{\mathrm{D}}_{m,k}}(\mathbf{b}^{\mathrm{D}}_{m,k}), \tag{3.19}$$

$$\bar{\mathbf{t}}^{\mathrm{I}}_{m-1,k} = \mathrm{c}^{-1}_{\psi^{\mathrm{I}}_{m,k}}(\mathbf{b}^{\mathrm{I}}_{m-1,k}), \tag{3.20}$$

where $\mathbf{b}^{\mathrm{I}}_{m-1,k}$ is the interfering CSI feedback received through the backhaul link. The output of the BS consists the precoding matrix $\mathbf{V}_m$ for the m-th cell

$$\mathbf{V}_m = \mathrm{G}_{\phi_m}\left[(\bar{\mathbf{t}}^{\mathrm{D}}_{m,1}, \ldots, \tilde{\mathbf{t}}^{\mathrm{D}}_{m,K}), (\bar{\mathbf{t}}^{\mathrm{I}}_{m-1,1}, \ldots, \tilde{\mathbf{t}}^{\mathrm{I}}_{m-1,K})\right], \tag{3.21}$$

where G_{ϕ_m} represents the m-th BS neural network with parameters ϕ_m. The output of each BS has to satisfy the power constraint by setting $\mathrm{Tr}(\mathbf{V}_m\mathbf{V}_m^H) = P$.

During training, in order to have an equivalent system with differentiable quantities, the entropy encoder-decoder pair is skipped, and the pseudo-quantized features are directly provided to the BS network, i.e.,

$$\mathbf{V}_m = \mathrm{G}_{\phi_m}\left[(\tilde{\mathbf{t}}^{\mathrm{D}}_{m,1}, \ldots, \tilde{\mathbf{t}}^{\mathrm{D}}_{m,K}), (\tilde{\mathbf{t}}^{\mathrm{I}}_{m-1,1}, \ldots, \tilde{\mathbf{t}}^{\mathrm{I}}_{m-1,K})\right]. \tag{3.22}$$

Note that during training, the full end-to-end optimization is equivalent to having a central processing unit, having access to all the CSI from all users in all cells, that seeks the precoding vectors that optimize the network rate (3.5). Instead, during testing the BS output (3.22) only depends on the CSI inputs from the same and neighboring cells, as in (3.10).

3.3.5 Loss Function

As discussed in Section 3.1, three metrics can be considered in the CSI feedback problem: feedback overhead, system performance, and channel distortion. In order to train the end-to-end system with deep learning techniques, we need a differentiable loss function that emulates the required properties for the system described in Section 3.2 and depicted in Fig. 3.3. We consider a loss function (to be minimized during training) that combines the three metrics as

$$\mathcal{L}(\Theta, \Psi, \phi) = \mathcal{O} - \lambda\mathcal{R} + \nu\mathcal{D}, \tag{3.23}$$

where $\mathcal{O}$ represents the feedback overhead, $\mathcal{R}$ represents the system performance (achievable rate), and $\mathcal{D}$ represents the distortion loss (channel reconstruction); λ and ν determine the tradeoff between the three components; $\Theta = \{\theta^{\mathrm{D}}, \theta^{\mathrm{I}}\}$, $\Psi = \{\psi^{\mathrm{D}}, \psi^{\mathrm{I}}\}$ and ϕ are the learnable parameters. We assume that the tradeoff coefficients are non-negative, i.e., $\lambda, \nu \geq 0$. For example, traditional overhead-distortion (or rate-distortion) settings correspond to $\lambda = 0$, while precoding-oriented systems correspond to $\nu = 0$. In our work, we will consider $\nu = 0$ and will sweep values of λ for different feedback overhead. Systems that provide both precoding vectors and channel reconstructions can be also considered by having non-zero values for both (λ, ν), but they are not the focus of this thesis. More details about the metrics are provided below.

3.3.5.1 Overhead

The feedback overhead accounts for the amount of bits that are required to transmit $(\mathbf{b}^{\mathrm{D}}_{m,k}, \mathbf{b}^{\mathrm{I}}_{m,k})$ on the uplink. As discussed previously, we use the empirical entropy of $\tilde{\mathbf{t}}^{\mathrm{D}}_{m,k}$ and $\tilde{\mathbf{t}}^{\mathrm{I}}_{m,k}$ as a measure for the feedback overhead [30]. Hence, we can express the overhead metric for the k-th user in the m-th cell as

$$\begin{aligned}\mathcal{O}_{m,k}(\Theta, \Psi) = \mathbb{E}_{\mathbf{h},\mathbf{g},\mathbf{u},\mathbf{z}}\big[&- \log_2 p_{\tilde{\mathbf{t}}}(\tilde{\mathbf{t}}^{\mathrm{D}}_{m,k}; \psi^{\mathrm{D}}) \\ &- \log_2 p_{\tilde{\mathbf{t}}}(\tilde{\mathbf{t}}^{\mathrm{I}}_{m,k}; \psi^{\mathrm{I}})\big],\end{aligned} \tag{3.24}$$

where $p_{\tilde{\mathbf{t}}}(\cdot; \psi)$ represents the approximated density of $\tilde{\mathbf{t}}$ parameterized by ψ. This loss term can be seen as an estimate for the number of bits required to represent the feedback vectors $(\mathbf{b}^{\mathrm{D}}_{m,k}, \mathbf{b}^{\mathrm{I}}_{m,k})$. The sum of the feedback overhead for the K users in the M cells can be expressed as

$$\mathcal{O}(\Theta, \Psi) = \sum_{m=1}^{M} \sum_{k=1}^{K} \mathcal{O}_{m,k}(\Theta, \Psi). \tag{3.25}$$

3.3.5.2 Performance

The system performance can be evaluated in terms of the achievable rate experienced by the users, as explained in Section 3.2. According to (3.4) and (3.5), we define the performance metric as

$$\mathcal{R}(\Theta, \Psi, \phi) = \sum_{m=1}^{M} \sum_{k=1}^{K} \mathcal{R}_{m,k}(\Theta, \Psi, \phi), \tag{3.26}$$

where

$$\mathcal{R}_{m,k}(\Theta, \Psi, \phi) = \mathbb{E}_{\mathbf{h},\mathbf{g},\mathbf{u},\mathbf{z}} \left[\log_2 \left(1 + \mathrm{SINR}_{m,k}\right)\right], \tag{3.27}$$

where $\mathrm{SINR}_{m,k}$ is defined in (3.3), and the precoding vectors $\mathbf{V}_{m,k}$'s are computed by the BS processing as in (3.10).

3.3.5.3 Distortion

Conventional methods (not precoding-oriented) rely on channel reconstructions to compute the precoding vectors with traditional algorithms. In these conventional methods, the CSI feedback is treated as a classic *rate-distortion* problem with $\lambda = 0$ and $\nu > 0$ in (3.23), and the mean squared error (MSE) is adopted as a distortion metric. We refer to this as a *reconstruction-oriented* CSI feedback, and the output of the BS processing are the estimates of the channel coefficients. Although this is not the focus of this thesis, in this case, the distortion component can be expressed as

$$\mathcal{D}(\Theta, \Psi, \phi) = \mathbb{E}_{\mathbf{h},\mathbf{g},\mathbf{u},\mathbf{z}} \left[||\mathbf{h}_{m,k} - \hat{\mathbf{h}}_{m,k}||^2 + ||\mathbf{g}_{m,k} - \hat{\mathbf{g}}_{m,k}||^2\right], \tag{3.28}$$

where $\hat{\mathbf{h}}_{m,k}$ and $\hat{\mathbf{g}}_{m,k}$ are channel reconstructions at the BS output when the CSI feedback is reconstruction-oriented.

3.3.6 Feedback Overhead Partitioning

Effective multi-cell precoding strategies require CSI for both the desired and interfering channels. Assuming that a fixed total overhead is available for the CSI feedback, the partitioning policy between desired and interfering CSI bits can improve the overall system performance. An analytical solution for the feedback partitioning in Rayleigh fading channels was proposed in [53]. In our work, since we adopt an end-to-end learned approach, we do not impose a structure on the feedback partitioning scheme. In fact, the overhead component (3.24) of our loss function (3.23) simply presents the sum of the overhead caused by the desired and interfering feedback. In case a specific partitioning policy is to be implemented, a scaling coefficient can be introduced for the two terms in (3.24) in order to give different importance to the desired and interfering feedback during training.

3.4 Results

We present simulation results for our precoding-oriented CSI feedback framework in three different scenarios: (i) single-cell multi-user; (ii) multi-cell single-user; (iii) multi-cell multi-user. Note that (i) exhibits intra-cell interference only, (ii) exhibits inter-cell interference only, and (ii) exhibits both intra- and inter-cell interference. These separate scenarios will allow us to better understand the overhead-performance tradeoffs and robustness in the following analysis. The system performance is evaluated as the sum achievable rate (3.26), while the feedback overhead (per user) is estimated according to (3.24).

First, we describe the simulation scenario and the channel data generation and define the neural network choices, the corresponding hyperparameters, and the training procedure. Then, simulation results for the three abovementioned scenarios are presented in separate subsections.

3.4.1 Simulation Scenario

While our framework is applicable to any channel model, we consider the following multipath channel model for our simulations, similar to [4]. We assume that each BS is equipped with a uniform linear array, with transmit array response

$$\mathbf{a}_{\mathrm{t}}(\beta) = \left[1, e^{j\frac{2\pi f d}{c}\sin(\beta)}, \ldots, e^{j\frac{2\pi f d}{c}(N_t-1)\sin(\beta)}\right], \tag{3.29}$$

where β denotes the angle of departure (AoD), d denotes the antenna spacing, f denotes the carrier frequency and c denotes the speed of light. The desired channel gains at the k-th user in the m-th cell are the summation of L_p propagation paths as

$$\mathbf{h}_{m,k} = \frac{1}{\sqrt{L_p^{\mathrm{D}}}} \sum_{\ell=1}^{L_p^{\mathrm{D}}} \alpha_{m,k,\ell}^{\mathrm{D}} \mathbf{a}_{\mathrm{t}}(\beta_{m,k,\ell}^{\mathrm{D}}), \tag{3.30}$$

where $\alpha_{m,k,\ell}$ is the complex gain of the ℓ-th path between the m-th BS and the k-th user. Without loss of generality, we assume that the interfering channel gains are characterized by a similar model

$$\mathbf{g}_{m,k} = \frac{1}{\sqrt{L_p^{\mathrm{I}}}} \sum_{\ell=1}^{L_p^{\mathrm{I}}} \alpha_{m,k,\ell}^{\mathrm{I}} \mathbf{a}_{\mathrm{t}}(\beta_{m,k,\ell}^{\mathrm{I}}). \tag{3.31}$$

We assume that each BS is equipped with $N_t = 64$ antennas, with spacing $d = c/(2f)$, and pilot duration $L = 8$. For the channel models (3.30) and (3.31), we assume that the number of paths is $L_p^{\mathrm{D}} = L_p^{\mathrm{I}} = 2$. The channel gains are i.i.d. $\alpha_{m,k,\ell}^{\mathrm{D}}, \alpha_{m,k,\ell}^{\mathrm{I}} \sim \mathcal{CN}(0,1)$. The desired AoD are i.i.d. $\beta_{m,k,\ell}^{\mathrm{D}} \sim \text{Uniform}[-60^o, 60^o]$ and the interfering AoD is i.i.d. $\beta_{m,k,\ell}^{\mathrm{I}} \sim \text{Uniform}[0^o, 120^o]$. The users' SNR is $\rho_{m,k} = \gamma_{m,k} P/\sigma_{m,k}$, where the downlink power constraint is $P = 1$, and we assume $\gamma = 1$ without loss of generality. The feedback link is noiseless, i.e., $(\mathbf{b}_{m,k}^{\mathrm{D}}, \mathbf{b}_{m,k}^{\mathrm{I}})$ is received at the BSs without distortion.

3.4.2 Neural Network Definitions and Training Procedure

As observed in [4, 47], using unique feedback schemes works well also in the multi-user scenario if the channel realizations have the same statistics according to the channel model. Unless otherwise specified, we assume symmetry in the channel conditions experienced by the users, so that we can consider a special case where the processing blocks are the same, i.e., $\mathcal{F}_{m,k} = \mathcal{F}$ and $\mathcal{G}_m = \mathcal{G}$, $\forall m, k$.

The neural networks and hyperparameters are chosen as follows. The pilots $\tilde{\mathbf{X}}_m \in \mathbb{C}^{N_t, L}$ are defined as learnable complex coefficients that satisfy the power constraint $||\tilde{\mathbf{x}}_{m,\ell}||^2 = P$. The user networks $\mathrm{F}_{\theta^{\mathrm{D}}}$ and $\mathrm{F}_{\theta^{\mathrm{I}}}$ consists of four fully-connected layers with hidden size $(5 \cdot N_b)$ and output size $N_b = 20$, while the BS network G_ϕ has five fully-connected layers with hidden size $(10 \cdot K \cdot N_t)$ and output size $(K \cdot N_t)$. Each hidden layer is preceded by batch normalization, and followed by a rectified linear unit (ReLU) activation. Real and imaginary parts of signals are processed on separate layers when appropriate. The last layers of both F_θ and G_ϕ have linear activation, For the feedback mechanism similar to [30], ψ parameterize the probability distribution of the feedback variable. We use the *entropy bottleneck* class from [57], which provides a PyTorch implementation of [58]. The end-to-end architecture is trained with ADAM optimizer, learning rate 10^{-3}, over at least 10^6 batches of size 1024. Unless otherwise specified, the parameters are initialized randomly. The numerical results are obtained on a test set containing at least 10^4 channel realizations for each user.

Training multi-cell multi-user systems can quickly become computationally challenging. For a large number of users, we suggest the following strategy to speed up the training procedure. First, the user networks are learned for a single-cell single-user system. Then, these pre-trained networks $(\mathcal{F}^{\mathrm{D}}, \mathcal{F}^{\mathrm{I}})$ can be used as an initialization or as fixed parameters (not further optimized) for the user networks. When the pre-trained user networks are used for initialization, we observed a faster convergence to good performance results. On the other hand, when these pre-trained networks $(\mathcal{F}^{\mathrm{D}}, \mathcal{F}^{\mathrm{I}})$ are fixed for each of the users in the cell, we can focus on fine-tuning only the parameters on the BS side (i.e., pilots $\tilde{X}$ and precoding processing $\mathcal{G}$). This is equivalent to a scenario where the mobile vendors have a fixed set of parameters (possibly more than one depending on location, CSI overhead target, etc.), and the BS stores different networks for handling different numbers of users and different scenarios.

3.4.3 Single-Cell Multi-User Scenario

We now discuss simulation results for the single-cell setting, where only the intra-cell interference component degrades the SINR (3.3). Since we are considering a single-cell scenario, the cell index m and the superscripts $^{\mathrm{D/I}}$ are omitted for brevity in this set of results. As reference precoding algorithms for the single-cell scenario, we also consider reconstruction-oriented systems that rely on traditional precoding schemes such as maximal-ratio transmission (MRT) and zero-forcing (ZF) [35, 59]. In MRT, the per-user received power is maximized, while ZF attempts to minimize the inter-user interference. Note that in the high (low) SNR asymptotic regime, the optimal linear precoding strategy converges to the ZF (MRT) solution [35]. In the perfect CSI at the transmitter (CSIT) scenario, each BS has perfect knowledge of all channel coefficients $\mathbf{H}$, where $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_K]$ is the matrix collecting the channel values for all of the K users in the cell. However, when a traditional reconstruction-oriented CSI feedback method is adopted, the BS has access to an estimate of the channel coefficients $\hat{\mathbf{H}}$; this is referred to as the imperfect CSIT scenario. The precoding matrices for MRT and ZF precoding are

$$\mathbf{V}_{\text{MRT}} = \tau_{\text{MRT}} \mathbf{A}^H \tag{3.32}$$

$$\mathbf{V}_{\text{ZF}} = \tau_{\text{ZF}} \mathbf{A}^H (\mathbf{A}\mathbf{A}^H)^{-1} \tag{3.33}$$

where τ_{MRT} and τ_{ZF} are determined to ensure that the power constraints $\text{Tr}(\mathbf{V}_{\text{MRT}}\mathbf{V}_{\text{MRT}}^H) \leq P$ and $\text{Tr}(\mathbf{V}_{\text{ZF}}\mathbf{V}_{\text{ZF}}^H) \leq P$ are satisfied; $\mathbf{A}$ corresponds to the true $\mathbf{H}$ in case of CSIT, or $\hat{\mathbf{H}}$ in case of imperfect CSIT.

Conventional methods rely on CSIT, by separating the source coding blocks (compressor and decompressor) from the task block (compute precoding). Note that previous works [26, 44, 47] showed that deep learning-based approaches outperform traditional techniques (e.g., compressed sensing) for the CSI feedback reconstruction problem. In particular, [47] showed that a deep learning-based CSI feedback architecture can be successfully trained to optimize the overhead-distortion tradeoff, when using a feedback optimization similar to the one described in Section 3.3.3; so we will consider the following deep learning-based approach as a surrogate for all conventional methods. To simulate this reconstruction-oriented approach, we set $\lambda = 0$ and $\nu > 0$ in our loss function (3.23), and consider $\hat{\mathbf{H}}$ as the output of the BS. Then, MRT and ZF precoders are computed using the channel estimates $\hat{\mathbf{H}}$ according to (3.32) and (3.33). The resulting precoding matrices are used to estimate the performance according to (3.26). We train the model for different values of ν to obtain neural networks with different overhead-distortion tradeoffs. For example, large (small) ν corresponds to a good (poor) reconstruction with a little (big) feedback overhead.

As another baseline algorithm, we also consider the best results in [4, Fig. 4 and 9], where the feedback is modeled as a vector of binary values. Each feedback overhead budget determines the dimension of the feedback in the architecture and the end-to-end system is trained to maximize the sum rate (3.26). Note that in this case the feedback overhead is not optimized explicitly during training.

Finally, we present the results of our precoding-oriented approach, which optimizes the overhead-performance tradeoff by training the end-to-end system with $\lambda > 0$ and $\nu = 0$ in the loss function (3.23). We recall that in this case the output of the BS are the precoding vectors for each user in the cell, and channel values are not explicitly reconstructed at the BS side. We obtain different overhead-performance tradeoff values by changing the value of λ: large (small) λ leads to good (poor) precoding performance with a small (large) feedback overhead. For this set of results, we assume that all the users have the same SNR $\rho = 10$ dB.

Fig. 3.4 shows the overhead-performance tradeoff for the above-mentioned methods in the simulation scenario described in Section 3.4.1. In addition, in order to compare with the previous literature, we consider AoD $\beta_{k,l} \sim \text{Unif.}[-30^o, 30^o]$. The SNR for all the users is set to $\rho = 10$ dB. Our precoding-oriented system

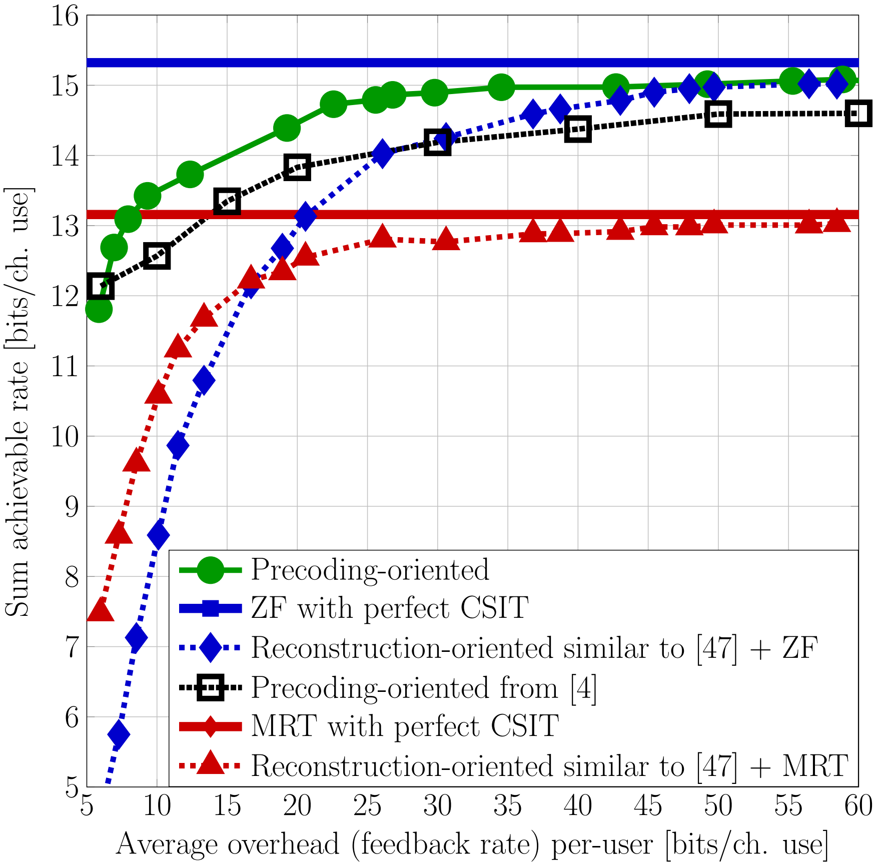


Figure 3.4: Analysis of the tradeoff between the feedback overhead and the system performance for a single-cell scenario ($M = 1$), with $K = 2$ users in the cell. Each marker corresponds to a different end-to-end architecture trained for different values of λ and ν in the loss function (3.23). In this case the angles are $\beta \sim \text{Unif.}[-30^o, 30^o]$ in order to compare with [4].

trained on the overhead-performance tradeoff is shown in green, and it outperforms all the other methods. In the large feedback overhead regime, the reconstruction-oriented system (similar to [47]) followed by ZF provides results comparable to our approach, making the two methodologies equivalent when considering the overhead-performance tradeoff. Our methods also provide a significant gain in performance compared to [4] (black line). This gain may be explained by the adaptability of our end-to-end solution, which includes a learning mechanism paired with entropy coding that directly accounts for the feedback overhead in the loss function, as explained in Section 3.3. Note that with our method the user's neural network is able to adapt to the overhead budget; i.e., the user is able to learn the efficient precoding-oriented feedback scheme. Compared with [4], our method is able to provide further compression using entropy coding. However, this requires an accurate representation of the distribution of the compressed variables, which our framework provides, as illustrated in Section 3.3. As expected, in the large feedback overhead regime, reconstruction followed by MRT/ZF approaches the performance of the perfect CSIT case. In fact, as the overhead increases, the BS is able to reconstruct the channel with decreasing distortion; hence, the traditional precoding algorithms can rely on more reliable channel estimates.

Fig. 3.5 shows the system performance as the number of users K increases in the single-cell system. To provide a comparison with [4], we consider channels with AoD $\beta \sim \text{Unif}[-60^o, 60^o]$, and feedback overhead per user of at most 30 bits/channel use. The SNR for all users is $\rho = 10$ dB. As previously noted, increasing the number of users K also increases the neural network size at the BS. With increasing K, the training becomes more challenging since more computation blocks are optimized jointly. In order to speed up training times, as explained in Section 3.4.2, we also provide results for a system where the: (a) the users are *pre-trained* from a single-cell single-user scenario, working at the target overhead of 30 bits/channel use; (b) with fixed pre-trained users, the BS is *fine-tuned* and trained to maximize sum-rate performance. As expected, since the users observe channel data from the same distribution, the performance of the systems trained for a specific K and $K = 1$ are equivalent. It can also be seen that our architecture optimized for the overhead-performance tradeoff (3.23) outperforms the previous work [4] that fixed

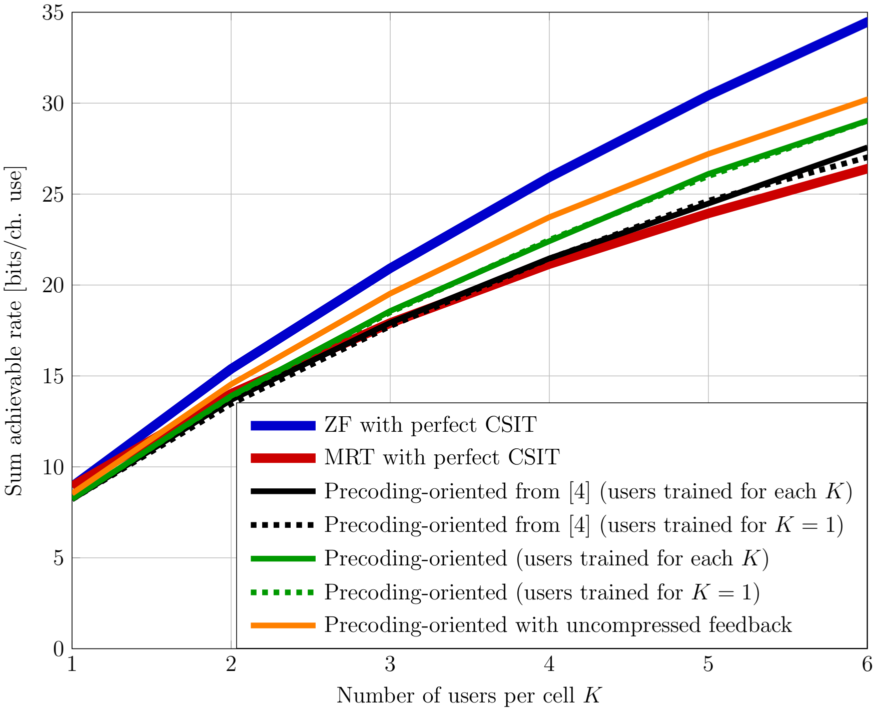


Figure 3.5: Sum rate performance achieved for a single-cell system ($M = 1$) with a variable number of users K. The average overhead per user is limited to at most 30 bits/channel use. The black and green dashed lines denote systems where only the BS in fine-tuned, and the users consist of pre-trained models learned for $K = 1$, while the solid counterparts are trained for each value of K. The orange line corresponds to a BS trained directly with the noisy received pilots, i.e., it models an uncompressed feedback. In this case the angles are $\beta \sim$ Unif.$[-60^o, 60^o]$ in order to compare with [4].

the feedback size and used a loss function based on the sum rate (3.5) only.

3.4.4 Multi-Cell Single-User Scenario

In this section, we present results for a multi-cell single-user scenario. We consider a system composed of $M = 10$ cells, each with $K = 1$ users (for a total of 10 users). In this case, the system exhibits inter-cell interference only. Note that although the system dimensions are similar to [52], in our setup we also include the downlink pilot training phase, and consider a different channel model. The system is evaluated in terms of the tradeoff between the sum rate performance (3.5) and the overhead required for the feedback of each user.

As reference baseline algorithms, we consider: (i) the perfect CSIT scenario, where the BSs have full access to both the desired channel gains $\mathbf{h}$ and the interfering channel gains $\mathbf{g}$; (ii) the uncompressed feedback scenario, where the users are bypassed, and the received pilots are directly fed to the BSs over a feedback link with infinite capacity (i.e., $\mathcal{O} \to \infty$ in (3.23)). Note that the perfect CSIT scenario bypasses both the downlink channel estimation phase (done with the pilots) and the CSI compression component, while the uncompressed feedback scenario skips only the latter.

Fig. 3.6 shows results for the multi-cell single-user scenario with $M = 2$ and $K = 1$, for different intra-cell interference ratios $\alpha \in \{0, 0.1, 1\}$. The solid lines represent the full end-to-end systems including pilots, user processing, and BS processing, where each marker denotes a different value of λ in (3.23). The annotations near each marker represent the resulting overhead split between desired

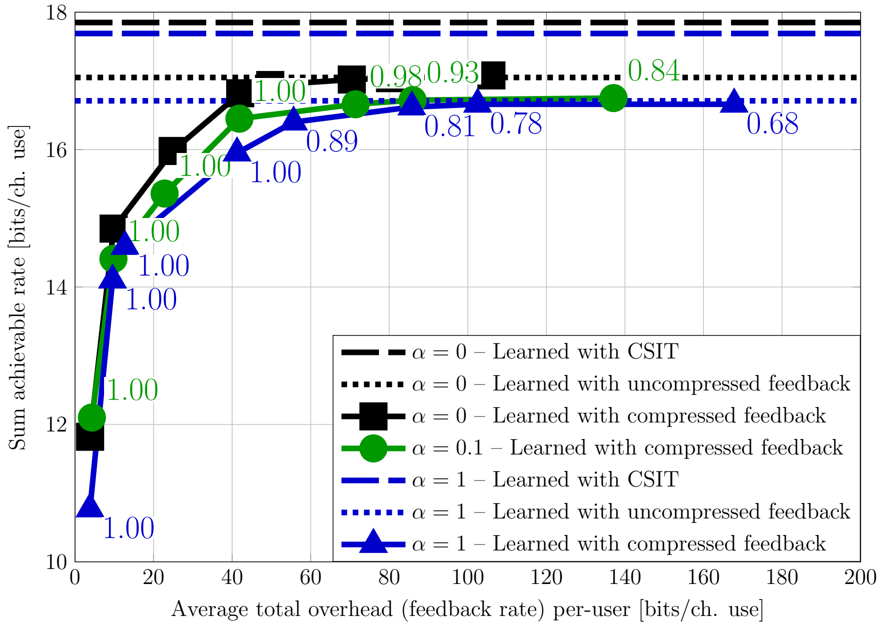


Figure 3.6: Analysis of the tradeoff between feedback overhead and system performance for the multi-cell single-user setting. The system includes $M = 2$ base stations, $K = 1$ users per cell (total of 2 users), and different interference ratios α. Each marker corresponds to an end-to-end architecture trained for a particular value of λ in (3.23). The text annotations near each marker represent the overhead split between desired and interfering CSI.

and interfering CSI, i.e.,

$$\text{split} = \frac{\text{desired CSI bits}}{\text{interfering CSI bits}}.$$

As expected, as α decreases, the system performance increases. Note that for $\alpha = 0$ (no interference), the performance is twice the single-cell single-user case (see Fig. 3.5 for $K = 1$) since this corresponds to $M = 2$ independent singe-user cells. Moreover, the split required to describe the desired channel is higher for lower interference α. We can also observe that in the low overhead regime (left region of the plot), the whole CSI overhead is dedicated to describing the desired channel; as the overhead increases (right region of the plot), the users allocate increasing

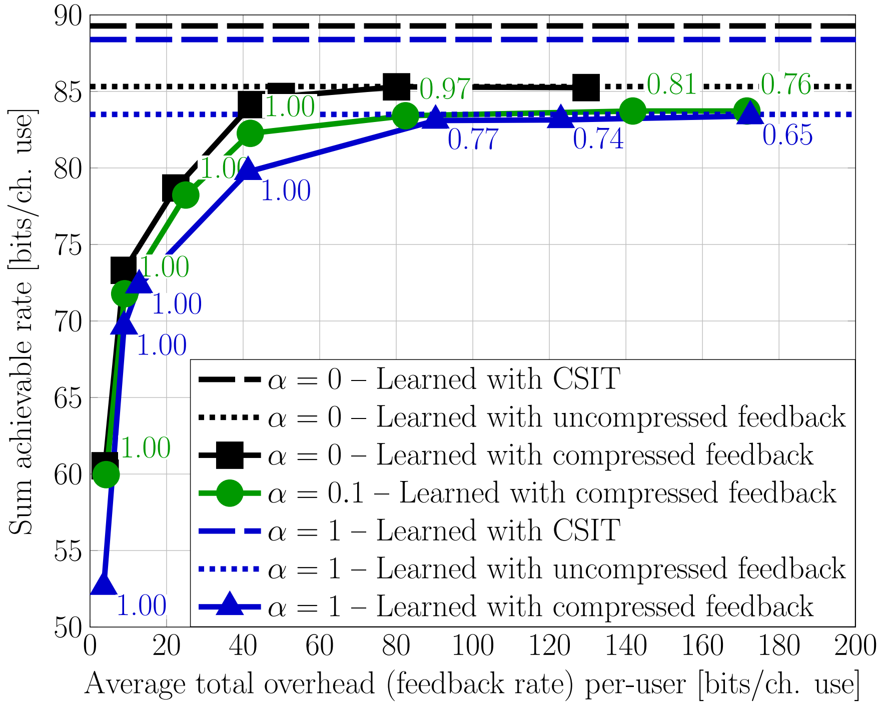


Figure 3.7: Analysis of the tradeoff between feedback overhead and system performance for the multi-cell single-user setting. The system includes $M = 10$ base stations, $K = 1$ users per cell (total of 10 users), and different interference ratios α. Each marker corresponds to an end-to-end architecture trained for a particular value of λ in (3.23). The text annotations near each marker represent the overhead split between desired and interfering CSI.

fractions of overhead to the interfering CSI.

To demonstrate the ability to handle larger networks, Fig. 3.7 shows results for the multi-cell single-user scenario with $M = 10$ and $K = 1$, for different inter-cell interference ratios $\alpha \in \{0, 0.1, 1\}$. We can see trends similar to the previous $M = 2$, $K = 1$ case. Note that for $\alpha = 0$, the performance is ten times the single-cell single-user case (see Fig. 3.5 for $K = 1$) since this corresponds to $M = 10$ independent singe-user cells in this case.

We can note that our performance tends to saturate to the uncompressed feedback regime for feedback overheads larger than N_t, where, as expected, higher

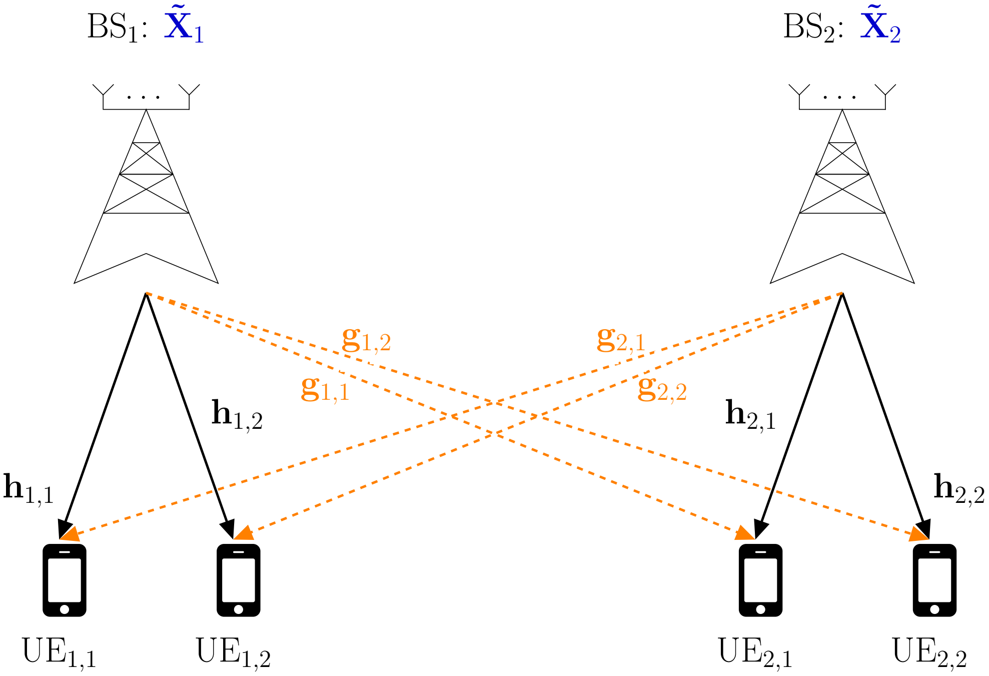


Figure 3.8: Downlink representation for the multi-cell multi-user simulation scenario with $M = 2$ base stations, $K = 2$ users per cell (total of 4 users). The solid black lines denote the desired channels $\mathbf{h}_{m,k}$, while the dashed orange lines denote the interfering channels $\mathbf{g}_{m,k}$. Each base station has its own set of pilots $\tilde{\mathbf{X}}_m$.

interference α requires more overhead. This indicates that excellent performance could be achieved with approximately 1 bit per antenna per channel use. In previous work [52] (which did not include the downlink pilots for channel estimation), performance saturates at a slower rate as the overhead increases [52, Cfr. Fig. 7, 8, 9], suggesting a less efficient precoding-oriented CSI feedback scheme.

3.4.5 Multi-Cell Multi-User Scenario

We now consider a fully multi-cell multi-user MIMO system with $M = 2$ BSs and $K = 2$ users per cell, for a total of 4 users in the system. A diagram of the considered scenario is shown in Fig. 3.8 (downlink), and in Fig. 3.9 (uplink). Note

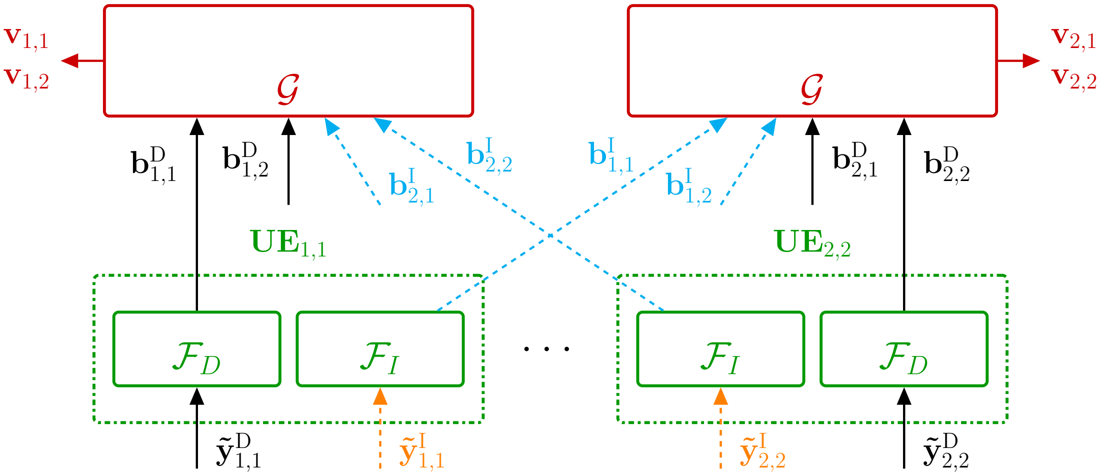

$\mathbf{v}_{1,1}$
$\mathbf{v}_{1,2}$
$\mathcal{G}$
$\mathcal{G}$
$\mathbf{v}_{2,1}$
$\mathbf{v}_{2,2}$
$\mathbf{b}^{\mathrm{D}}_{1,1}$
$\mathbf{b}^{\mathrm{D}}_{1,2}$
$\mathbf{b}^{\mathrm{I}}_{2,1}$
$\mathbf{b}^{\mathrm{I}}_{2,2}$
$\mathbf{b}^{\mathrm{I}}_{1,1}$
$\mathbf{b}^{\mathrm{I}}_{1,2}$
$\mathbf{b}^{\mathrm{D}}_{2,1}$
$\mathbf{b}^{\mathrm{D}}_{2,2}$
$\mathbf{UE}_{1,1}$
$\mathbf{UE}_{2,2}$
$\mathcal{F}_D$
$\mathcal{F}_I$
$\cdots$
$\mathcal{F}_I$
$\mathcal{F}_D$
$\tilde{\mathbf{y}}^{\mathrm{D}}_{1,1}$
$\tilde{\mathbf{y}}^{\mathrm{I}}_{1,1}$
$\tilde{\mathbf{y}}^{\mathrm{I}}_{2,2}$
$\tilde{\mathbf{y}}^{\mathrm{D}}_{2,2}$

Figure 3.9: Uplink representation for the multi-cell multi-user simulation scenario with $M = 2$ base stations, $K = 2$ users per cell (total of 4 users). The feedback $\mathbf{b}^{\mathrm{I}}_{m,k}$ denoted in cyan are received by the base station through the backhaul link.

that in this case, both the intra-cell and inter-cell interference phenomena degrade the SINR (3.3) at the users.

Fig. 3.10 shows the overhead-performance tradeoff for the multi-cell multi-user scenario with $M = 2$ and $K = 2$, for different intra-cell interference levels α. Note that for $\alpha = 0$ (no interference), the performance is twice the single-cell single-user case (see Fig. 3.5 for $K = 2$) since this corresponds to $M = 2$ independent two-user cells. Also for this setting, as expected, the performance improves as the interference α decreases. Each marker corresponds to a specific value of λ in (3.23) and the annotations denote the ratio of overhead that is spent on the desired feedback w.r.t. the total overhead. Similar to the multi-cell single-user setting, we observe that in the low overhead regime, the whole CSI overhead is dedicated to the desired channel.

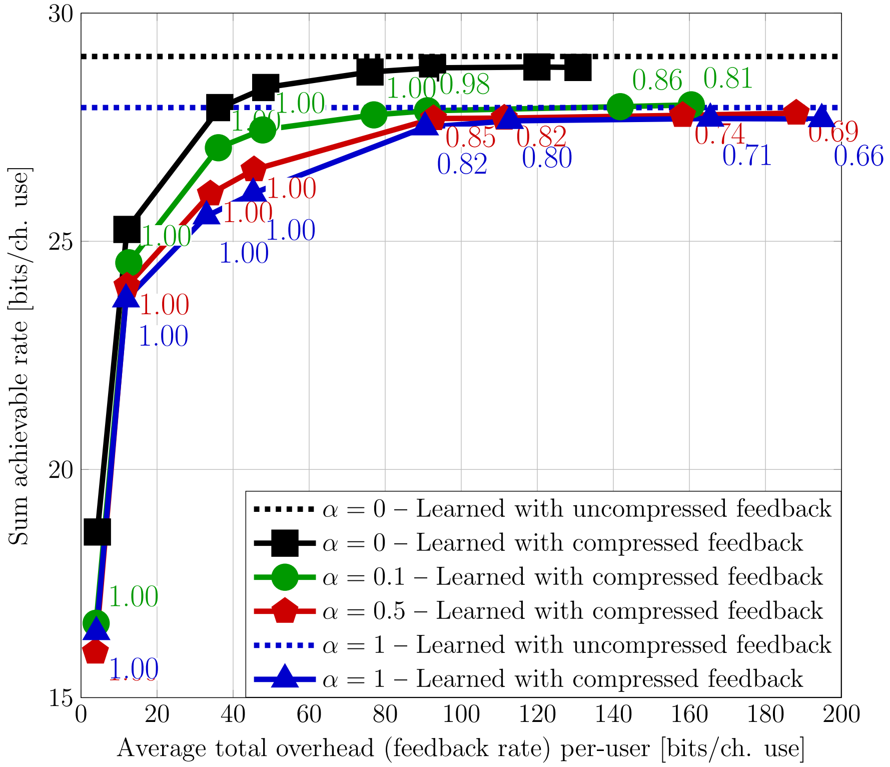
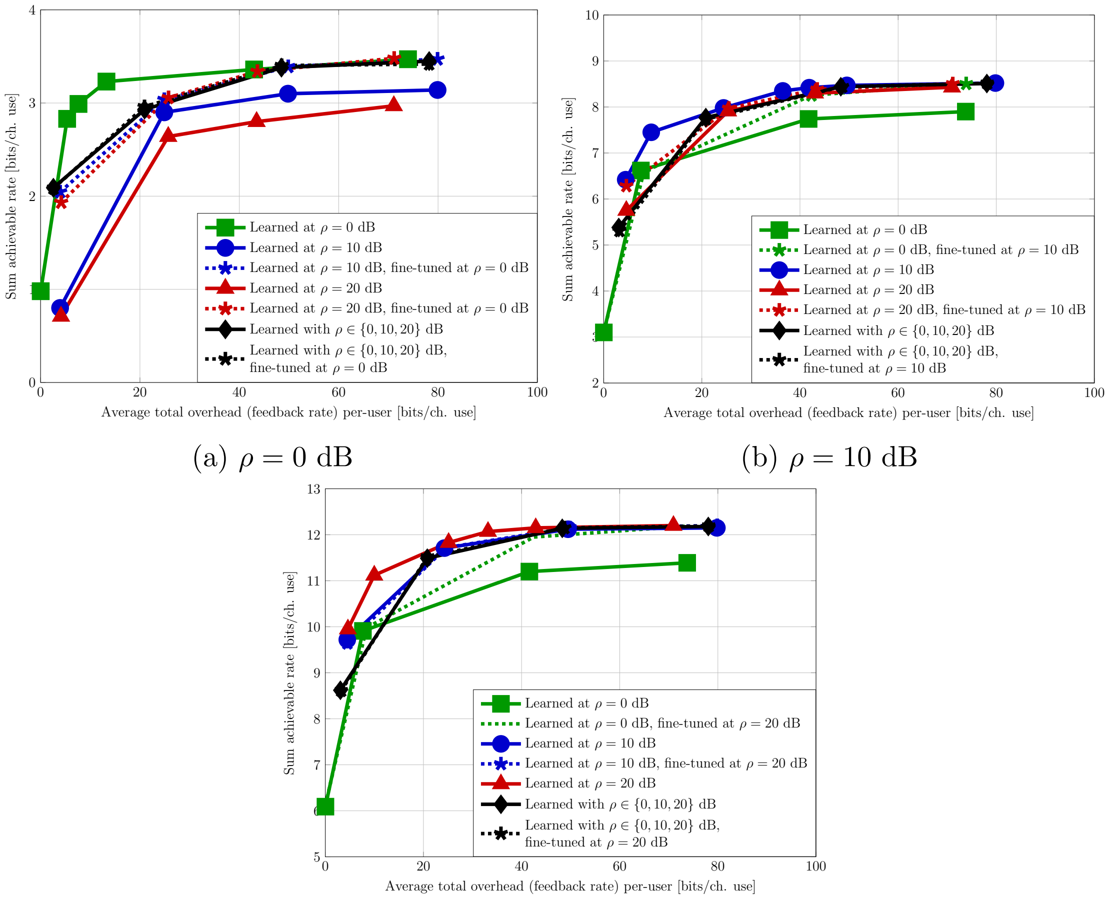


Figure 3.10: Analysis of the tradeoff between feedback overhead and system performance for the multi-cell multi-user setting. The system includes $M = 2$ base stations, $K = 2$ users per cell (total of 2 users), and different interference ratios α. Each marker corresponds to an end-to-end architecture trained for a particular value of λ in (3.23). The text annotations near each marker represent the overhead split between desired and interfering CSI.

3.4.6 Robustness to Different SNR

A key aspect in transitioning from proof-of-concepts to practical systems is ensuring robustness across diverse scenarios. In this section, we address this challenge by examining the mismatch between training and testing SNRs ρ at different users. For these experiments, we suppose that the SNR is $\rho \in \{0, 10, 20\}$ dB. We consider models trained at a single SNR, and over a range of SNR $\rho \sim$ Unif$\{0, 10, 20\}$ dB. Moreover, we also consider the same models after fine-tuning only the BS side for the target test SNR. This last scenario corresponds to the



(a) $\rho = 0$ dB

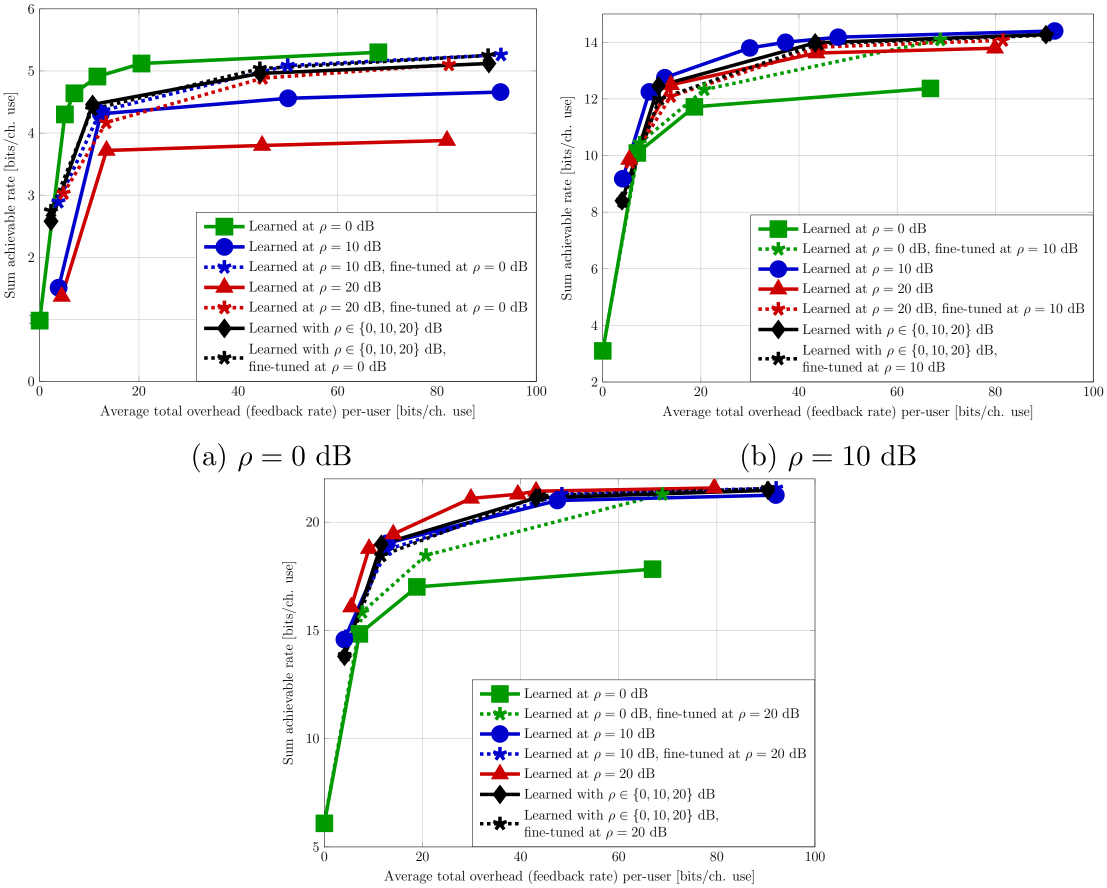
(b) $\rho = 10$ dB

(c) $\rho = 20$ dB

Figure 3.11: Analysis for the single-user scenario without interference, i.e., $M = 1$, $K = 1$, $\alpha = 0$, for different SNR ρ. Each marker corresponds to an end-to-end architecture trained for a particular value of λ in (3.23).

assumption that the BS can have multiple models to pick from, while the users may have one (or few) models that they can use.

Fig. 3.11 analyzes the robustness of the SNR for a single-user system. For large overheads, fine-tuning improves the performance and all the fine-tuned models match the best performance for each SNR. For small overheads, the models trained at $\rho = 0$ dB do not improve after fine-tuning. On the other hand, the models trained at the higher SNRs $\rho = 10, 20$ dB benefit from tuning when tested for $\rho = 0$



(a) $\rho = 0$ dB

(b) $\rho = 10$ dB

(c) $\rho = 20$ dB

Figure 3.12: Analysis for the two-user scenario without interference, i.e., $M = 1$, $K = 2$, $\alpha = 0$, for different SNR ρ. Each marker corresponds to an end-to-end architecture trained for a particular value of λ in (3.23).

dB. Moreover, for the higher SNRs the learned models for $\rho = 10, 20$ perform well even without fine-tuning. The model trained over the range $\rho \in \{0, 10, 20\}$ dB provides a good compromise if the users are SNR-agnostic.

Fig. 3.12 analyzes the robustness of the SNR for a two-user system with the same SNR $\rho_1 = \rho_2 = \rho$. For large overheads, fine-tuning improves the performance and reduces the gap with the best-performing model for each SNR. We observe similar behaviors as in the single-user case, where, in general, models trained at

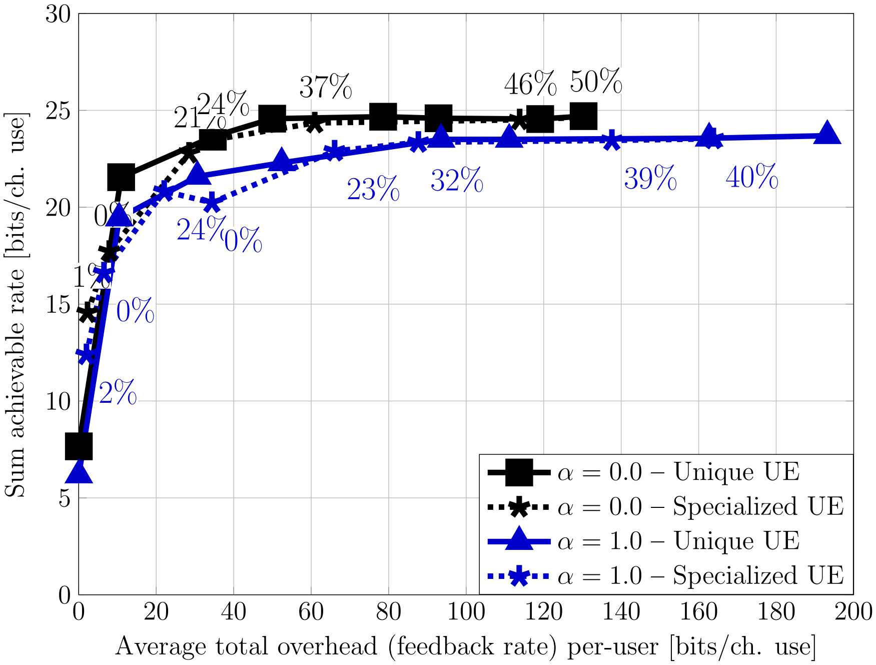


Figure 3.13: Analysis of the tradeoff between feedback overhead and system performance for the multi-cell multi-user setting. The system includes $M = 2$ base stations, $K = 2$ users per cell (total of 2 users), and different interference ratios α. User 1 in each cell has an SNR of $\rho_{m,1} = 2$, while user 2 has $\rho_{m,2} = 18$. The solid lines represent systems where all of the users share the same neural network, while the users' have individual (specialized) neural networks in the dashed lines. The annotation denotes the percentage of overhead that is used by the user 1.

high SNR exhibit better generalization properties. Again, if the operating SNR is unknown to the users, the model trained over the range $\rho \in \{0, 10, 20\}$ dB gives satisfactory performance.

Fig. 3.13 analyzes the robustness with respect to different SNR for a two-cell two-user system (4 users total). We assume that the SNR for the first user in each cell is $\rho_{m,1} = 3$ dB, and the SNR for the second user in each cell is $\rho_{m,2} = 12.55$ dB. The solid lines denotes systems where all the users share the same neural networks, while the dashed lines represent systems where each user has its own learned model. The annotation near the dashed lines denotes the percentage of overhead that is used by the first user in each cell – that is 50% for the solid lines since the users' models are shared and work at the same CSI bitrate.

3.5 Summary

In this thesis, we have analyzed the tradeoff between feedback overhead, performance, and distortion in the CSI feedback problem for multi-cell multi-user massive MIMO systems operating in FDD. We have investigated the effects of intra- and inter-cell interference in the design of the task-aware CSI strategy. We also addressed a robustness issue, comparing performance when the SNR at the users does not match the one used for training. Our proposed deep learning-based precoding-oriented CSI feedback mechanism demonstrates flexibility in the learning part, thanks to an ad-hoc loss function that captures the key performance indicators and it is adaptable to any neural architecture. In general, we observed that by defining an unstructured loss function (3.23) (i.e., without giving weights specific to the channel conditions), the system learns a policy that recalls water-filling. As the overhead budget increases, the system first allocates feedback bits to the *strong* components (i.e., CSI of the desired channel and/or user with better SNR), then starts allocating overhead to the *weaker* components (i.e., CSI of interfering channel and/or user with worse SNR).

Some future directions for this thesis include the extensions to system-level simulations, with state-of-the-art channel models (e.g., ray tracing-based), where the entire end-to-end communication stack is optimized jointly. Moreover, the extension to the MIMO-OFDM framework, which includes the frequency dimension of the channel gains, is also an interesting direction.

Chapter 4

Detection-Oriented Relays

The relay channel, consisting of a source-destination pair along with a relay, is a fundamental component of cooperative communications. While the capacity of a general relay channel remains unknown, various relaying strategies, including compress-and-forward (CF), have been proposed. In CF, the relay forwards a quantized version of its received signal to the destination. Given the correlated signals at the relay and destination, distributed compression techniques, such as Wyner–Ziv coding, can be harnessed to utilize the relay-to-destination link more efficiently. Leveraging recent advances in neural network-based distributed compression, we revisit the relay channel problem and integrate a learned task-aware Wyner–Ziv compressor into a primitive relay channel with a finite-capacity out-of-band relay-to-destination link. The resulting neural CF scheme demonstrates that our compressor recovers binning of the quantized indices at the relay, mimicking the optimal asymptotic CF strategy, although no structure exploiting the knowledge of source statistics was imposed into the design. The proposed neural CF, employing

This chapter refers to a joint work with Ezgi Ozyilkan, and it will be presented at IEEE SPAWC 2024 [60].

finite order modulation, operates closely to the rate achievable in a primitive relay channel with a Gaussian codebook. We showcase the advantages of exploiting the correlated destination signal for relay compression through various neural CF architectures that involve end-to-end training of the compressor and the demodulator components. Our learned task-oriented compressors provide the first proof-of-concept work toward interpretable and practical neural CF relaying schemes.

4.1 Introduction

The relay channel, as introduced by van der Meulen [61], is a building block of multi-user communications. In this model, a relay facilitates communication between a source and a destination by forwarding its “overheard” received signal to the destination. As such, the relay channel comprises a *broadcast channel*, from the source to both the relay and the destination, and a *multiple access channel*, from both the source and the relay to the destination. The relay channel forms the foundation of cooperative networking, which has been shown to be effective in mitigating fading [62, 63], increasing data rates [64], and managing interference [65]. With the advent of 6G, new forms of relaying and cooperation are envisioned for communicating in highly dynamic settings [66, 67].

Despite decades of research, the capacity of the general relay channel is still unknown to this day. Cover and El Gamal [68] provided upper and lower bounds for the general relay channel by invoking information theoretic achievability and converse arguments. These bounds coincide only in a few special cases, such as the physically degraded Gaussian relay channel. Even though optimum relaying strategies are not known in general, various effective relaying techniques have been

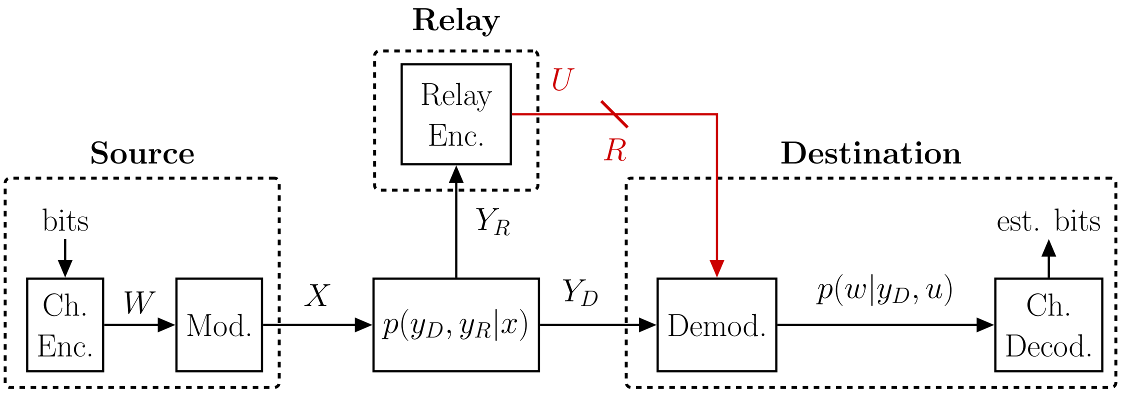


Figure 4.1: The *primitive* relay channel (PRC) under consideration. The red link denotes out-of-band relaying between the relay and the destination.

proposed, which can be broadly categorized into two main classes: *decode-and-forward* (DF) and *compress-and-forward* (CF); see [68] for a detailed analysis of DF, CF, their variations and combinations. While DF is known to be efficient in certain scenarios [64], its achievable rate is bounded by the capacity of the source-to-relay channel since the relay is required to perfectly decode the source information.

On the other hand, in CF, the relay refrains from directly decoding the source and instead, compresses its received signal to send to the destination. Upon reception of the compression index, the destination combines it with its own received signal to decode the source information. Given that the received signals at the relay and destination are correlated, the relay can leverage *distributed compression* techniques to reduce the compression rate without requiring explicit knowledge of the received signal at the destination. As such, it can utilize Wyner–Ziv (WZ) source coding [69], also known as source coding with decoder-only side information, to efficiently describe its received signal. Unlike DF, CF relaying consistently outperforms direct transmission since the relay always aids in communication, even when the source-to-relay channel is poor. For additional discussion on scenarios where CF has been proven to be optimal, we direct readers to [70]. Despite its benefits, the limitations of practical WZ implementations operating in the finite blocklength regime have hampered the widespread use of CF relaying.

In this work, drawing on recent advances in neural distributed compression [71, 72], we revisit practical CF design and illustrate the potentials of learning for reaping the benefits of CF. To highlight design constraints for CF, we focus on the *primitive* relay channel (PRC) [73], depicted in Fig. 4.1, where there is an orthogonal (out-of-band) noiseless link of rate R connecting the relay to the destination. Our main contributions are summarized as follows:

- We present learned CF relaying schemes for the Gaussian PRC that are based on task-aware neural distributed compressors, where the task is to maximize the source-to-destination communication rate. We provide several architectures, differing in the way the distributed compression is carried out. Each of these schemes consists of a compressor at the relay and (soft) demodulator at the destination, both of which are learned in an end-to-end fashion.

- We offer post-hoc interpretations of the resulting neural CF schemes on some representative modulation schemes. Through visualizations, we show that the task-aware neural relay quantizer exhibits *binning* (grouping) in the source space, which is known to be information theoretically optimal. In addition, we illustrate explainable decision boundaries for the learned demodulator at the destination. These structures emerge from learning, not from design choices based on system parameters.

- Using a comprehensive set of experimental results, we evaluate the performance of our neural CF strategies both in terms of communication and error rates. Comparison with theoretical benchmarks suggests the effectiveness of our learning-based relaying frameworks.

- We provide a detailed analysis of robustness to varying signal-to-noise ratios (SNRs) both at the relay and the destination. We empirically demonstrate that training over a range of SNRs enables the resulting CF strategy to maintain good performance across the range of interest.

Overall, our learned CF framework represents the first proof-of-concept investigation towards practical and robust CF relaying, with the added benefit of yielding interpretable results.

A few comments are in order regarding our motivation for considering the PRC. Firstly, the PRC offers a scenario where the compressed relay signal can be readily transmitted to the destination. Compared to the general relay channel, the PRC model decouples the relay transmission from that of the source, allowing a natural setting to study CF. Note that the PRC model represents the simplest channel coding problem, viewed from the source's perspective, with a rate constraint among the two receiving terminals (relay and destination). Simultaneously, it also encapsulates the simplest compression problem, viewed from the relay's perspective, for enabling channel coding between the source and the destination. Secondly, PRC provides a good model for scenarios in which a different wireless or wired interface is used for relaying, such as base station cooperation. Finally, the relaying strategies developed for the PRC can be extended to a more general relay channel model by incorporating the multi-access reception at the destination. We also note that CF relaying is optimal for the PRC if the relay is unaware of the source codebook, also known as *oblivious* relaying [5]. The oblivious setting is well-suited to the learning framework, in which the relay is not explicitly informed about the transmission strategy used by the source. Rather, a data-driven relay trains its compressor based on samples of its channel output.

There is limited literature addressing practical CF designs, e.g., [74, 75]. Both of these works proposed entropy-constrained scalar quantizer designs with binary phase shift keying (BPSK) modulation for the half-duplex Gaussian relay channel, with [74] considering lossless Slepian–Wolf (SW) coded nested quantization as a practical form of WZ compression (following the WZ compressor proposed in [76]), and [75] not taking into account the side information at the destination while quantizing at the relay. In addition, these works relied on handcrafted and analytical solutions, thereby constraining their generalization to more complex communication settings. Unlike some of the previous distributed compression work (e.g., [76, 77]) or the aforementioned relay quantizer designs, our proposed CF strategies neither enforce any specific structure onto the model nor assume prior knowledge about the source-to-destination communication strategies or link qualities.

Recent learning approaches for the relay channel [78, 79, 80] considered a joint source-channel setting, where the first two focused on image transmission via joint source-channel coding, while the last one targeted text communication utilizing attention-based transformer architectures. Our work, in contrast, concentrates *only on the channel* part and addresses an important open problem in the cooperative communications literature, namely how to make CF practical. While our learned CF framework is built upon those of [71, 81, 82], an important distinction is that in CF, the goal is to facilitate source-to-destination communication, and not to reconstruct the relay signal per se. In fact, it is demonstrated in [75] that relay compression that minimizes mean squared error distortion can be significantly suboptimal. We refer the reader to [72] for an overview of distributed compression and practical designs, including those based on neural networks, that focus on signal reconstruction.

This section is organized as follows. The system model is explained in Sec. 4.2. The proposed neural CF schemes and learning procedures are described in Sec. 4.3. Extensive numerical results are presented in Sec. 4.4. A concluding summary and future work are discussed in Sec. 4.5.

4.2 System Model

In this section, we introduce the PRC model (Fig. 4.1) and provide an achievable rate for CF, which is tight for oblivious relaying. Next, we explain the performance criterion we adopt for testing relaying schemes that involve task-aware neural distributed compressors.

4.2.1 Primitive Relay Channel (PRC)

We consider the PRC setup [73], illustrated in Fig. 4.1. The Gaussian PRC, which we study in this work, is given by:

$$\begin{aligned} Y_R &= h_R\, X + N_R, \\ Y_D &= h_D\, X + N_D, \end{aligned} \tag{4.1}$$

where X denotes the signal transmitted by the source, Y_R and Y_D denote the received signals at the relay and the destination, and h_R and h_D are the corresponding channel gains, respectively. The noise components, N_R and N_D, are independent of one another and of X.

In this work, we consider both real and complex-valued channels. For the real-valued channel, without loss of generality, we consider $X, h_R, h_D \in \mathbb{R}$, $N_R \sim \mathcal{N}(0,1)$ and $N_D \sim \mathcal{N}(0,1)$. For the complex-valued channel, we assume $X, h_R, h_D \in \mathbb{C}$, $N_R \sim \mathcal{CN}(0,1)$ and $N_D \sim \mathcal{CN}(0,1)$. Note that by allowing for arbitrary (h_R, h_D), one can incorporate the effect of different SNRs for the source-to-relay and source-to-destination links. As customary, we consider communication over a blocklength of n, with n asymptotically large, and i.i.d. noise. For brevity, we omit the time index in (4.1). The out-of-band relay-to-destination channel is represented by a link with *relay rate* R bits/channel use.

For a general PRC $p(y_D, y_R|x)$ with an oblivious relay, where the relay is agnostic to the codebook shared by source and destination, it was shown that the capacity can be attained by the CF strategy with time sharing [5]. Without time-sharing, the following rate C is achievable [5]:

$$C = \max \ \mathrm{I}(X; Y_D, U), \tag{4.2}$$

$$\text{s.t.} \ \ R \geq \mathrm{I}(Y_R; U \mid Y_D), \tag{4.3}$$

where maximization is with respect to the distribution $p(x)p\left(u|y_R\right)$. Here, U corresponds to the relay's compressed description of Y_R, and the rate constraint in (4.3) coincides with the one that emerges in WZ rate–distortion function [69]. Recall that in CF, the relay regards its received signal Y_R as an unstructured random process jointly distributed with the signal received at the destination Y_D. This enables the relay to exploit WZ compression [69], to efficiently describe its received signal. We note that the capacity of the PRC without oblivious relaying constraint is still not fully characterized [5].

For the real-valued Gaussian PRC in (4.1), the following CF rate is achieved with Gaussian input under power constraint $\mathbb{E}[|X|^2] \leq P$ [5]:

$$C_{\mathrm{CF}} = \frac{1}{2}\log_2\left(1 + \gamma_D + \frac{\gamma_R}{1 + \frac{1+\gamma_D+\gamma_R}{(2^{2R}-1)(\gamma_D+1)}}\right), \tag{4.4}$$

where $\gamma_D = |h_D|^2 P$ and $\gamma_R = |h_R|^2 P$ are SNRs at the destination and at the relay, respectively. Note that, in the case of a complex-valued PRC, the factor of 1/2 in (4.4) is removed. It is shown in [5] that while the Gaussian input is not necessarily optimal, the rate in (4.4) is at most 1/2 bit away from the capacity of the Gaussian PRC, even if the relay is not oblivious. Hence, we will use (4.4) as a benchmark for our learned CF communication rates.

4.2.2 Performance Criterion

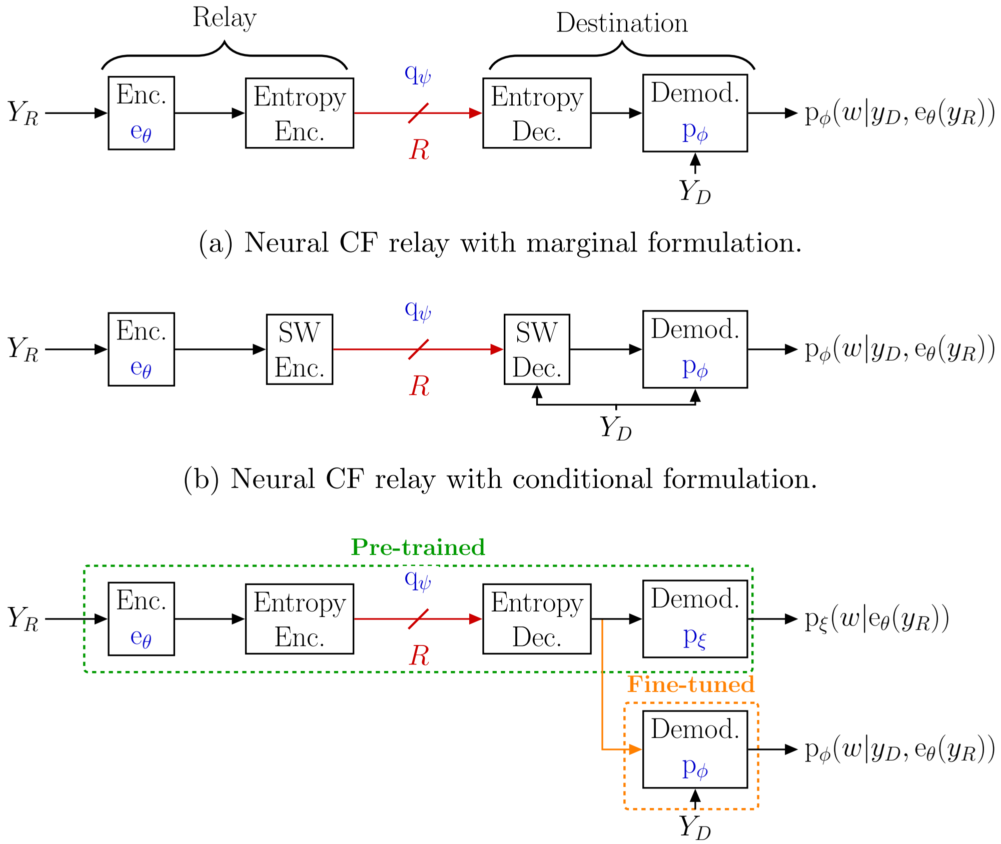
For our learning-based CF frameworks, we assume a finite order modulation such that an index $W \in \{1, \ldots, |\mathcal{X}|\}$, which represents the output of the channel encoder, is mapped to a symbol $X \in \mathcal{X}$, where $\mathcal{X} \subset \mathbb{R}$ (or, $\mathcal{X} \subset \mathbb{C}$ for complex-valued signals) is a constellation of cardinality $|\mathcal{X}|$. We consider a fixed modulation scheme with equally likely symbols, and do not optimize over the constellation $\mathcal{X}$ or over the distribution $p(x)$. Incorporating the learned probabilistic and geometric constellation shaping [83] into our neural CF frameworks is beyond the scope of this work. Our goal is to jointly learn the *encoder* at the relay, which outputs a compressed description U, and the (soft) *demodulator* at the destination, which outputs a probability distribution on W (Fig. 4.1) that maximize the mutual information $\mathrm{I}(X; Y_D, U)$ subject to the relay rate constraint R, as in (4.2) and (4.3). We assume the availability of good channel codes to be used in conjunction with the modulation scheme, and as such the mutual information $\mathrm{I}(X; Y_D, U)$ can be viewed as a CF achievable rate. In Sec. 4.3.3, we will discuss how this performance criterion is incorporated into the objective function used in the learning process.

4.3 Neural Compress-And-Forward (CF) Schemes

In this section, by leveraging universal function approximation capability of artificial neural networks (ANNs) [84, 85], we propose three neural CF schemes to be employed in the PRC shown in Fig. 4.1. We describe these schemes in detail in Sec. 4.3.1 and provide design insights in Sec. 4.3.2. Objective function and implementation details are discussed in Sec. 4.3.3. As detailed in in Sec. 4.2.2, the modulation scheme remains fixed throughout. On the other hand, the relay's encoder, employing CF strategy, and the destination's demodulator will be parameterized using ANNs, which will undergo joint optimization in an end-to-end manner.

4.3.1 Neural CF Architectures

Building onto neural distributed compressors proposed in [71], we consider learning-based CF schemes that include neural one-shot WZ compressors (with side information Y_D at the destination), paired with either a classic entropy coder (EC) or a SW coder, at the relay. We will name these two variants as *marginal* (marg.) and *conditional* (cond.) formulations, respectively. As a benchmark, we



(a) Neural CF relay with marginal formulation.

(b) Neural CF relay with conditional formulation.

(c) Neural CF relay with p2p formulation.

Figure 4.2: The three proposed neural CF schemes: (a) and (b) are based on marginal (*marg.*) and conditional (*cond.*) formulations, (coupled with classic either entropy or Slepian-Wolf (SW) coder) respectively; (c) is the point-to-point (*p2p*) scheme. The learned parameters are indicated in blue. Note that the schemes in (a) and (b) operationally correspond to task-aware neural Wyner–Ziv compressors, since the encoder can exploit the side information Y_D at the receiver side. In (c), neither parameters of e_θ and q_ψ are updated during the fine-tuning step (only p_ϕ is learned). In the *split I-Q* variants of each scheme (not depicted), we have two separate encoders that compress in-phase and quadrature components of the complex-valued signal independently. Wherever we present relevant experiments, we label the depicted respective scheme illustrated in this figure as *joint I-Q*, indicating a single encoder for both in-phase and quadrature components.

also consider a neural one-shot *point-to-point* (p2p) compressor coupled with a classic EC. All of these learned compressors are combined with a neural demodulator available at the destination, which has access to the side information Y_D.

The overall proposed learned CF relaying architectures are illustrated in Fig. 4.2. The encoder's ANN at the relay is denoted by $\mathrm{e}_\theta(\cdot)$, with θ representing its parameters; the probability distribution of the relay encoder's output (which is then used by the EC or SW coder) is modeled with q_ψ, parameterized by ζ; the demodulator's ANN is $\mathrm{p}_\phi(w|y_D, \mathrm{e}_\theta(y_R))$, where ϕ denotes its parameters. The mapping defined by the demodulator p_ϕ represents the posterior probability over the alphabet $\{1, \ldots, |\mathcal{X}|\}$ (soft decision), which serves as an approximation of the true posterior distribution $p(w|y_D, \mathrm{e}_\theta(y_R))$. In the learning process of a point-to-point compressor, as shown in Fig. 4.2c, we initially train a demodulator $\mathrm{p}_\xi(w|\mathrm{e}_\theta(y_R))$ to prevent this neural compressor from utilizing the side information Y_D during training. The pre-trained point-to-point neural compressor as such (highlighted in green) is then used as input for fine-tuning the demodulator $\mathrm{p}_\phi(w|y_D, \mathrm{e}_\theta(y_R))$, which incorporates side information (highlighted in orange).

We set the relay encoder's output as $U \triangleq \mathrm{e}_\theta(Y_R)$ as in Fig. 4.1. Envisioning a practical scheme, we have U as discrete. Specifically, we have that $\mathrm{e}_\theta(Y_R) \in \{1, ..., K\}$, where K is a model parameter. This parameter K is chosen large enough to guarantee sufficient support for the encoder output. To facilitate the learning process of the encoder, we will use a probabilistic model for $\mathrm{e}_\theta(Y_R)$ during training. We set the encoder output in a deterministic way, as in [71], that is $u = \arg\max_{k \in \{1,\ldots,K\}} \mathrm{e}_\theta(y_R)$ for a given $Y_R = y_R$. Note that the encoder e_θ operates in an unordered *categorical* space, outputting one of the categories of the quantization index $k \in \{1, \ldots, K\}$ for each input realization.

Similar to [71], without loss of generality, we define the probabilistic models $\mathrm{e}_\theta(Y_R)$ (during training) and q_ψ as discrete distributions with probabilities as follows:

$$P_k = \frac{\exp \alpha_k}{\sum_{i=1}^{K} \exp \alpha_i}, \tag{4.5}$$

for $k \in \{1, \ldots, K\}$. The unnormalized log-probabilities (*logits*) α_i are either directly treated as learnable parameters or computed by ANNs as functions of the conditioning variable. We note that the lossless compression rates induced by the models q_ψ are attainable with high-order classic EC [86] or SW coder [87], operating on discrete values.

For experiments involving complex-valued modulation schemes, the CF architectures depicted in Fig. 4.2 compress the in-phase (i.e. real) and quadrature (i.e. imaginary) components of Y_R *jointly.* We also explore the variants of these architectures illustrated in Fig. 4.2 (not shown), where the in-phase and quadrature components are given as input to two separate encoders, each of which has parameters of its own. The compression rate in this case is computed as the sum of the rates achieved by the in-phase and the quadrature entropy coding schemes (involving either classic EC or SW coders for both). Similar to the architectures outlined in Fig. 4.2, we still employ a single demodulator p_ϕ for all these variants, which takes as input the two indices coming from the compressed representations of the in-phase and the quadrature components, along with the side information Y_D. We will refer to these architectural configurations as the *split I-Q* variants, whereas we will name the original architectures shown in Fig. 4.2 as the *joint I-Q* versions.

Although joint compression of in-phase and quadrature components using schemes illustrated in Fig. 4.2 should, in principle, outperform independent processing as in the split I-Q variant, we argue that incorporating domain knowledge into the design, especially when training learning-based schemes, can sometimes facilitate finding the optimal solution for the algorithm. This will be further clarified in

Sec. 4.4.2.

We refer the readers to Sec. 4.3.3 for further details on the training procedure.

4.3.2 Rationale Behind Our Design Choices

While the popular class of neural image compressors (e.g., [88, 89, 90]) seems well-suited for distributed compression, and more specifically for the WZ problem, analysis in [82] reveals that it fails to learn efficient many-to-one mappings exploiting the side information. Consequently, these popular schemes do not recover proper binning schemes, which are known to be optimal in the asymptotic setting [69], for abstract exemplary sources (such as the quadratic-Gaussian case), severely limiting their compression efficiency. In [82], it is hypothesized that this limitation stems from the inherent *spectral bias* [91] of the popular class of neural compressors. This spectral bias arises because the encoder outputs operate on the real line. This inherently favors learning smooth functions, consequently hindering these neural compressors from capturing highly discontinuous functions and many-to-one mappings such as binning.

Based on this, our proposed learning-based CF schemes, as in the case of the learned WZ compressors [71, 82], operate directly within an unordered *categorical* space, similar to traditional vector quantization. Our neural relay compressors are, therefore, in the form of entropy-constrained vector quantizers that can more easily leverage correlated signal available at the destination. Note that this is in contrast to the popular class of neural compressors [88, 89, 90], where each of the dimensions at the encoder output is subjected to entropy-constrained scalar quantization in an ordered transform space operating on real line.

The design choices explained in Sec. 4.3.1 maintain the parametric families in their most general form, avoiding any unnecessary imposition of structure. In particular, these would enable the model e_θ to recover, when necessary, quantization schemes featuring discontiguous quantization bins, reminiscent of the *random binning* operation in the achievability of the WZ theorem [69], which also appears in the CF relaying strategy [5, 68].

4.3.3 Objective Function

In contrast to prior works on neural distributed compression [71], which focus on minimizing the *distortion* in the reconstruction of the input source in tandem with variable rate entropy coding, our goal in this work is to optimize the operational trade-off between relay-to-destination compression rate and source-to-destination communication rate in the PRC setup, underscoring the task-aware nature of the relay compressor design.

For our objective function, building onto the relay rate in (4.3), we first consider the following upper bound:

$$\begin{aligned} \mathrm{I}(Y_R; U \mid Y_D) &\leq H(U \mid Y_D), && (4.6) \\ &\leq \mathbb{E}\left[-\log_2 \mathrm{q}_\psi(\mathrm{e}_\theta(y_R))\right] \triangleq \tilde{R}, && (4.7) \end{aligned}$$

where $\tilde{R}$ represents an operational upper bound on the relay's *compression* rate, which is limited by R. The inequality in (4.7) is due to the fact that the cross-entropy is larger or equal to entropy [41, Theorem 5.4.3]. Here, $\tilde{R}$ encapsulates the compression rate of a relay quantizer having a one-shot encoder coupled with high-order entropy coder over large blocks of the quantized source.

Similarly, we also establish a lower bound based on the achievable rate in (4.2) as follows:

$$I(X; Y_D, U) = H(W) - H(W \mid Y_D, U), \tag{4.8}$$

$$\geq \log(|\mathcal{X}|) - \tilde{D}, \tag{4.9}$$

where $\tilde{D} \triangleq \mathbb{E}\left[-\log(\mathrm{p}_\phi(x|y_D, \mathrm{e}_\theta(y_R)))\right]$, and (4.9) is a lower bound on the source-to-destination *communication* rate C from (4.2). Here, (4.8) follows from X being a one-to-one deterministic function of W, and (4.9) is again due to cross-entropy being larger or equal to entropy. Since we have a fixed modulation scheme and do not perform any probabilistic shaping, we have $H(W) = H(X) = \log(|\mathcal{X}|)$ in (4.9).

For a demodulator making hard decisions as:

$$\hat{W} = \arg \max_{w \in \{1, \ldots, |\mathcal{X}|\}} \mathrm{p}_\phi(w|y_D, \mathrm{e}_\theta(y_R)), \tag{4.10}$$

the corresponding symbol error rate (SER) is defined as:

$$\mathrm{SER} = P(W \neq \hat{W}). \tag{4.11}$$

Since minimizing the cross-entropy $\tilde{D}$ is known to be a surrogate for maximizing the accuracy of classification (that is symbol detection) [92], minimizing $\tilde{D}$ also operationally corresponds to minimizing SER.

Building onto the above bounds, the training objective of all the proposed neural CF relaying schemes depicted in Fig. 4.2 can be described by the following loss function:

$$L(\theta, \phi, \zeta) = \tilde{R} + \lambda \tilde{D}, \tag{4.12}$$

where $\tilde{R}$ and $\tilde{D}$ are from (4.7) and (4.9) respectively, and $\lambda > 0$ controls the trade-off. The optimized e_θ, q_ψ, and p_ϕ models, parameterized by θ, ζ and ϕ, yield the ANN-based encoder, EC or SW coder, and demodulator component, respectively. The upper bound in (4.7) corresponds to the compression rate achievable by a CF relaying scheme employing a one-shot task-aware encoder e_θ and demodulator p_ϕ, both coupled with an entropy code based on q_ψ (either classic EC or SW coder). This asymptotic compression rate is equivalent to the cross-entropy $\mathbb{E}\left[-\log_2 \mathrm{q}_\psi(\mathrm{e}_\theta(y_R))\right]$. Similarly, the lower bound in (4.9) corresponds to the overall communication rate achieved by a capacity-achieving channel code, operating over large blocklengths, used in conjunction with the (soft) demodulator p_ϕ. Therefore, minimizing the loss function in (4.12) enables the end-to-end optimization of this *operational* relaying scheme.

Consistent with findings in [74, 75], we empirically confirmed that minimizing mean squared error distortion metric at the quantizers may not always maximize the source-to-destination communication rate. The intuition for this is as follows: A distortion-minimizing quantizer aims to preserve *the relay's received signal*, whereas the relay quantizer should instead retain the *source information* as the relay's end goal is to facilitate the communication in the source-to-destination link, highlighting the task-aware nature of the compressor design objective at hand. Grounded in information theoretical principles following [5], this key insight underlies the objective function (see (4.12)) for our neural CF relaying schemes.

Adjusting the trade-off parameter λ in (4.12) results in different points within the achievable region. The learnable parameters are amenable to joint optimization using stochastic gradient descent (SGD) since the loss function is differentiable with respect to them. The gradients can be computed using automatic differentiation methods, as implemented in deep learning frameworks such as JAX [93].

As in the popular class of neural compressors [90], we use SGD to optimize all learnable parameters jointly, which relies on Monte Carlo approximation for the expectations in the loss function. In SGD, the expectations in the loss functions are replaced by averages over batches of samples B, and the order of differentiation and summation is exchanged due to linearity. For a given generic pair of $X = x$ and $Y = y$, let $\ell_\theta(x, y)$ denote the sample loss with parameters θ (represented as one of the sample loss functions inside the brackets in (4.12)). In this case, Monte Carlo approximation yields:

$$\frac{\partial}{\partial\theta}\mathbb{E}[\ \ell_\theta(x, y))\ ] \approx \frac{1}{|B|}\sum_{(x,y)\in B}\frac{\partial \ell_\theta(x, y)}{\partial\theta}\ . \tag{4.13}$$

This requires that we draw some samples from the model e_θ throughout training. The Gumbel-max trick, initially proposed in [94], provides a method to draw samples from any discrete distribution. It does so by drawing samples from a distribution of K states (as in (4.5)) as follows:

$$\arg\max_{k\in\{1,\ldots,K\}}\{\alpha_k + G_k\}, \tag{4.14}$$

where G_k are i.i.d. samples from a standard Gumbel distribution.

Recognizing that the derivative of the $\arg\max$ operator in (4.14) is zero everywhere except at the boundaries of state changes, we opt for a *continuous relaxation* of this operator during training to carry out SGD. Such a relaxation is provided by the Concrete distribution, introduced in [95]. Rather than obtaining discrete (hard) samples, this method produces soft samples, forming a vector of length K where the mass is distributed across multiple states instead of being concentrated in one. The index $k \in \{1, \ldots, K\}$ of such a soft sample is determined using a *softmax* function:

$$U_k = \frac{\exp((\alpha_k + G_k) \;/\; t)}{\sum_{i=1}^{K} \exp((\alpha_i + G_i) \;/\; t)} \;, \tag{4.15}$$

where t is a temperature parameter that controls the amount of relaxation. As $t \to 0^+$, the soft samples converge to their hard counterparts, indicating that the Concrete distribution converges to a discrete one. Throughout training, we also choose the Concrete distribution for the models q_ψ to match the distribution of samples from e_θ.

During evaluation, we transition from Concrete distributions back to their discrete counterparts. As explained in Sec. 4.3.1, we also use a deterministic encoding function equivalent to the mode of e_θ, instead of sampling from it, by setting encoder output as $u = \arg\max_{k\in\{1,\ldots,K\}} \mathrm{e}_\theta(y_R)$.

Note that in spite of considering specific modulation schemes in training, we do not assume *a priori* knowledge of the modulation scheme by the relay in our neural CF schemes. The parameters $\{\theta, \phi, \zeta\}$ are learned solely in a data-driven fashion from samples, through the proposed loss function in (4.12). Similarly, the relay also has no prior information on the channel gains h_R and h_D (see (4.1)). Further improvement in the performance may be obtained by also learning an optimized probabilistic shaping ($p(x)$ in optimization (4.2)-(4.3)) and a geometric shaping (constellation $\mathcal{X}$) of the modulation [83].

4.4 Results and Discussion

While our framework can be adapted to different modulation schemes and PRC setups, we adopt the following system configuration to showcase numerical results. As stated in Sec. 4.2, we assume equally likely symbols, i.e., $p(x) = 1/|\mathcal{X}|$. The average power constraint on the transmitted signal is $\mathbb{E}[|X|^2] = P$. For real-valued channels, we consider BPSK, 4-PAM, and 8-PAM modulations, having constellations $\mathcal{X} = \{\pm A\}$, $\mathcal{X} = \{\pm A, \pm 3A\}$, and $\mathcal{X} = \{\pm A, \pm 3A, \pm 5A, \pm 7A\}$, respectively, where A is chosen to satisfy the power constraint P. For complex-valued channels, we consider 4-QAM and 16-QAM modulations with power constraint P. We recall that the SNR at the destination and at the relay is defined as $\gamma_D = |h_D|^2 P$ and $\gamma_R = |h_R|^2 P$, respectively.

For the parametrization of e_θ and p_ϕ, we use ANNs of three dense layers, with 100 units each, except the last one, and leaky rectified linear unit as the activation function. In our experiments, we observed that increasing the size of the networks or employing different activation functions did not lead to improved results. The demodulator p_ϕ receives a concatenated vector comprising both its inputs, $\mathrm{e}_\theta(Y_R)$ and Y_D.

We perform our experiments using the JAX framework [93] and employ Adam [96], a widely used variant of SGD. We use a learning rate of 10^{-4}, which we chose by monitoring the convergence of the loss function in the high-rate regime, reducing it by a constant factor of 10 each time the loss visibly plateaued. We found that convergence of the high-rate models takes longer than low-rate models, so we simply carried over our schedule to the lower-rate cases. All neural CF schemes are trained for 500 epochs with randomly initialized network weights. We use a batch size of $B = 1024$ (as in (4.13)) and set the model parameter $K = 32$. The output dimension of p_ϕ is set to be $|\mathcal{X}|$, since this probabilistic model represents the posterior over the transmitted constellation.

We evaluate our learned CF relaying schemes in terms of the trade-off between the relay rate R (using the proxy $\tilde{R}$ in (4.7)), and two metrics: (i) the communication rate $\mathrm{I}(X; Y_D, U)$, for which we use the lower bound (hence, a pessimistic estimate) in (4.9), and (ii) the SER $= P(W \neq \hat{W})$ (see (4.11)). All empirical estimates of compression rates, communication rates, and bit error rates are obtained by averaging over at least 10^6 source realizations.

The rest of this section is organized as follows. Baseline references for $R = 0$ and $R \to \infty$ are presented in 4.4.1. The performance of various learned CF relay schemes is analyzed in 4.4.2, while an interpretation of the corresponding relay's encoder and destination's demodulator is provided in 4.4.3. Finally, results for robustness against different SNRs are shown in 4.4.4.

4.4.1 Baselines

The regimes where $R = 0$ and $R \to \infty$ are referred to as *without relay* and *perfect relay* scenario, respectively. When $R = 0$, the destination has only access to Y_D, having an effective SNR of γ_D. In the perfect relay regime ($R \to \infty$), however, the destination has full access to both Y_D and Y_R, and it optimally combines them. This effectively results in an increased SNR of $\gamma_D + \gamma_R$ compared to the scenario without a relay. In these two regimes, mutual information and SER can be numerically computed for the considered modulations as a function of (γ_D, γ_R) [97].

When $0 < R < \infty$, we consider C_{CF} from (4.4) (or its complex channel equivalent) as a benchmark for the achievable communication rate of our learned CF schemes with discrete modulations. Increasing the modulation order, $|\mathcal{X}|$, gives more degrees of freedom for the end-to-end learned communication system to approach the rate of a PRC that assumes Gaussian inputs, as represented by C_{CF} in (4.4).

4.4.2 Performance of the Learned CF Relaying Schemes

For the first set of results, we assume that the SNR is the same for both the destination and the relay, i.e., $\gamma_D = \gamma_R$.

Fig. 4.3 shows the SER and mutual information for the 4-PAM modulation when $\gamma_D = \gamma_R = 13$ dB. In this case, Y_R and Y_D, are highly correlated. We observe that the three models exhibit different trade-offs. Recall that, in the point-to-point variant depicted in Fig. 4.2c, e_θ is not able to use Y_D as side information in compression. As seen in Fig. 4.3, the conditional model yields the best performance as the side information is also exploited within the SW coder. The marginal model surpasses the point-to-point model mainly due to exploiting the side information during compression (see Sect. 4.4.3 for a more detailed discussion), yielding rate reduction.

Similarly, Fig. 4.4 contains the SER and the mutual information for 16-QAM modulation when $\gamma_D = \gamma_R = 7$ dB. In this case, we also show results where two separate encoders compress the in-phase and quadrature part of Y_R independently – these models are annotated as *split I-Q* variants, as introduced in Sec. 4.3.1. We observe that at lower rates, the architectures depicted in Fig. 4.2, which correspond

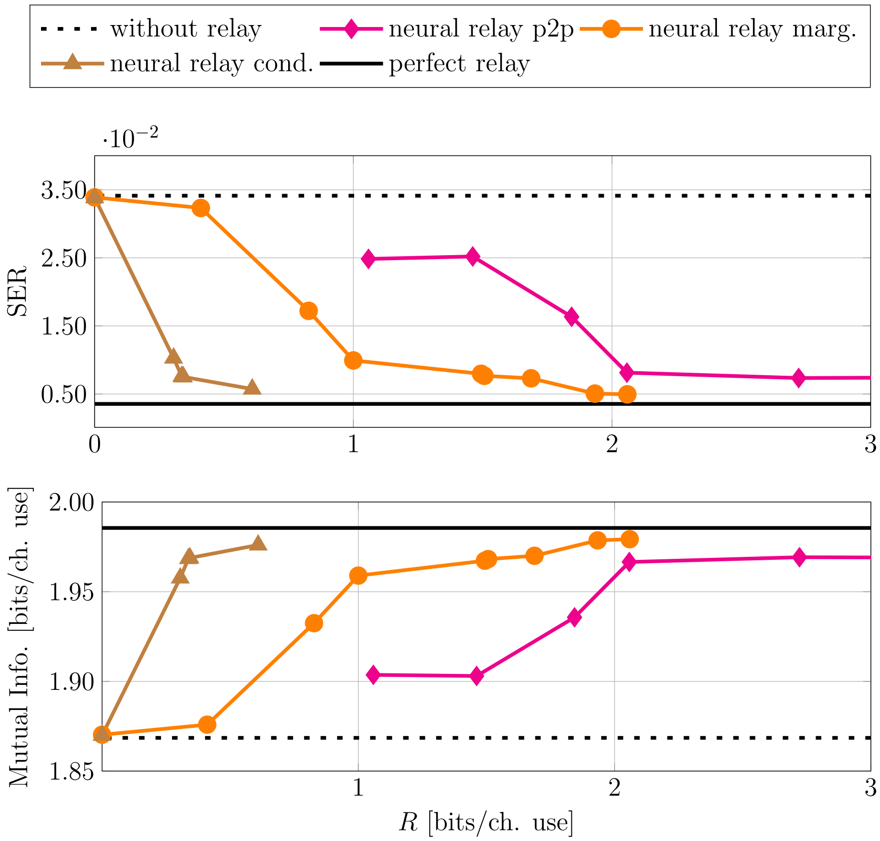


Figure 4.3: Symbol error rate (SER) and mutual information as a function of the relay-to-destination rate R, for the 4-PAM modulation with $\gamma_D = \gamma_R = 13$ dB. The colored lines represent the performance of three neural CF relay architectures (Fig. 4.2), where each marker corresponds to a unique model trained for a particular value of λ in (4.12). The horizontal black lines provide baseline results without relaying ($R = 0$) and with perfect relaying ($R \to \infty$).

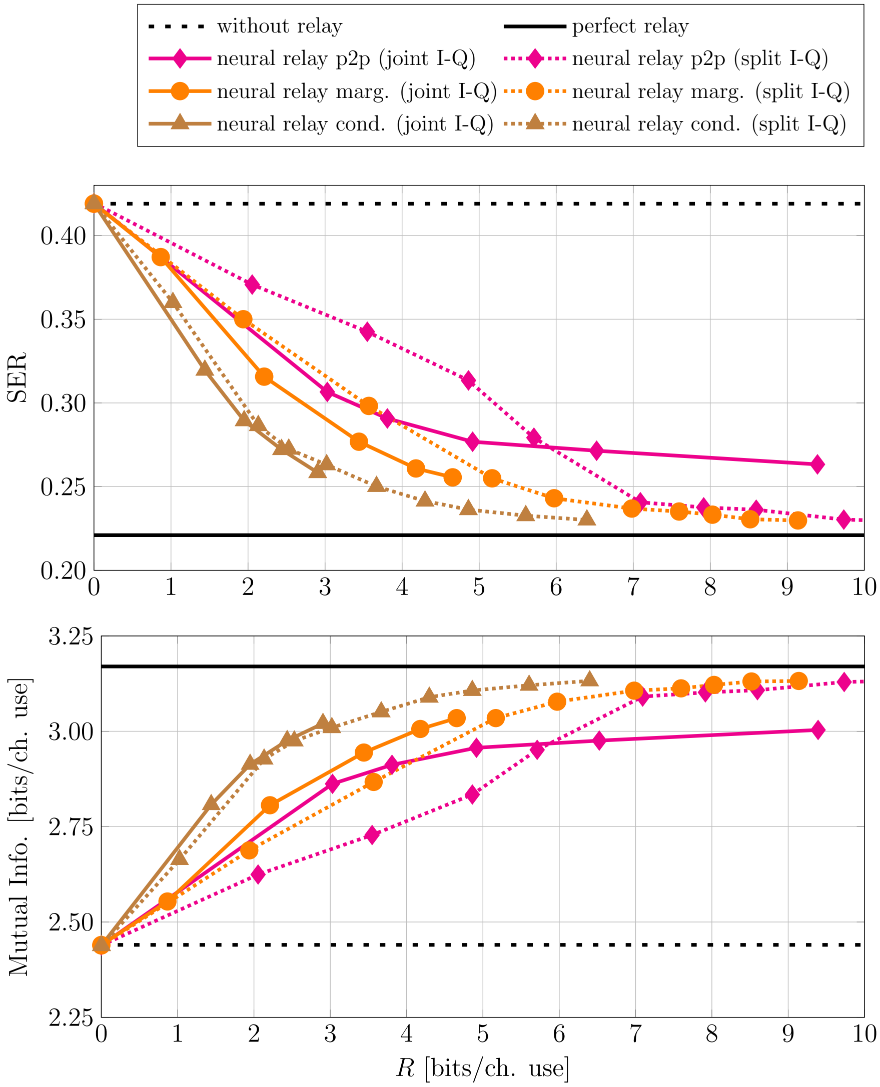


Figure 4.4: Symbol error rate (SER) and mutual information as a function of the relay-to-destination rate R, for the 16-QAM modulation with $\gamma_D = \gamma_R = 7$ dB. The colored lines illustrate the performance of three neural CF relay architectures depicted in Fig. 4.2, accompanied by their respective *split I-Q* variants (as introduced in Sec. 4.3.1). In the figure, each marker corresponds to a unique model trained for a specific value of λ in (4.12). The horizontal black lines indicate baseline results without relaying ($R = 0$) and with perfect relaying ($R \to \infty$).

to *joint I-Q* compression, perform best across three different schemes (conditional, marginal and point-to-point). In these models, both the in-phase and quadrature components are fed into a single encoder e_θ, enabling joint compression of real and imaginary parts of the complex-valued input signal. This allows these compressors to learn more flexible quantization boundaries (not depicted), making them more efficient in the low-rate regime. However, at higher rates, we observe that the split I-Q variants outperform their joint counterparts. Since one would expect "grid-like" quantization boundaries for QAM modulations, imposing separate processing on real and imaginary parts in the split I-Q models, effectively leverages this domain knowledge, enabling them to approach capacity at high rates. In contrast, all of the joint I-Q architectures for conditional, marginal and point-to-point variants saturate around a capacity value of 3. These results suggest that as the modulation order and relay rate increase, incorporating domain knowledge into the compressor design could be beneficial. Imposing such well-informed design structures in this case further enhances the efficiency of learned CF relaying schemes, particularly at high rates, where training neural compressors becomes relatively more challenging compared to the low rate regime.

Fig. 4.5 compares C_{CF} from (4.4) with the mutual information obtained with the marginal formulation (Fig. 4.2a) for the BPSK, 4-PAM and 8-PAM modulations. Here, the SNR for all the considered schemes is $\gamma_D = \gamma_R = 3$ dB, suggesting a lower correlation between Y_R and Y_D compared to the one illustrated in Fig. 4.3. As expected, increasing the modulation order narrows the gap to the bound in (4.4). Notably, the marginal variant meets the performance of the corresponding perfect relay ($R \to \infty$) baseline at higher rates.

Similarly, Fig. 4.6 compares C_{CF} from (4.4) for the same modulation schemes

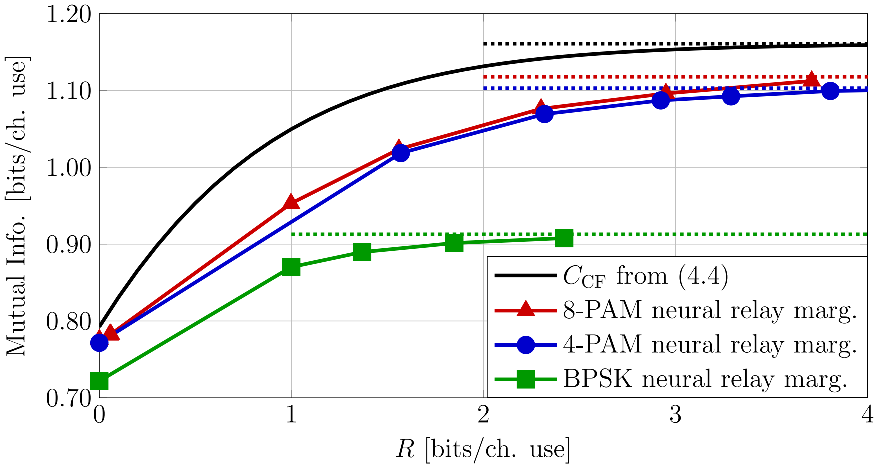


Figure 4.5: Mutual information for the marginal model (Fig. 4.2a) in case of BPSK, 4-PAM and 8-PAM modulations with $\gamma_D = \gamma_R = 3$ dB. The solid line represents C_{CF} in (4.4) [5], obtained for Gaussian inputs. The dotted lines represent the perfect relay ($R \to \infty$) bounds for the respective curves, similar to Figs. 4.3 and 4.4.

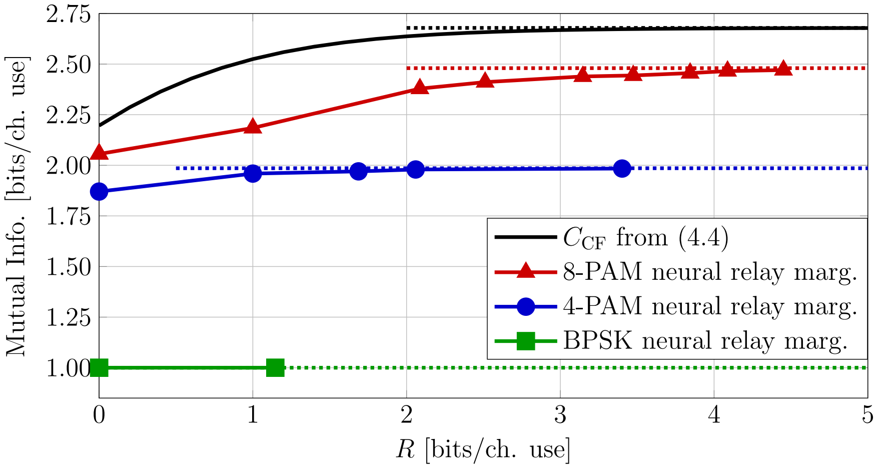


Figure 4.6: Mutual information for the marginal model (Fig. 4.2a) in case of BPSK, 4-PAM and 8-PAM modulations with $\gamma_D = \gamma_R = 13$ dB. The solid line represents C_{CF} in (4.4) [5], obtained for Gaussian inputs. The dotted lines represent the perfect relay ($R \to \infty$) bounds for the respective curves, similar to Figs. 4.3, and 4.4.

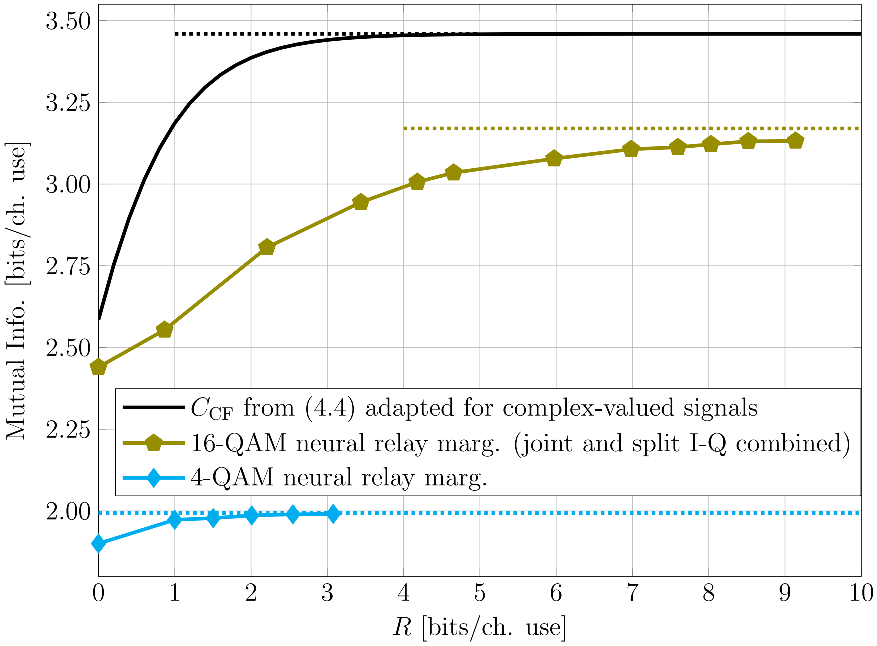


Figure 4.7: Mutual information for the marginal model (Fig. 4.2a) in case of 4-QAM and 16-QAM modulations with $\gamma_D = \gamma_R = 7$ dB. The solid line represents C_{CF} in (4.4) [5], obtained for Gaussian inputs. The dotted lines represent the perfect relay ($R \to \infty$) bounds for the respective curves, similar to Figs. 4.3 and 4.4. In the case of the 16-QAM modulation shown here, we select the best performing model among two variants of the marginal neural relay quantizers: the joint I-Q scheme depicted in Fig. 4.2a and the split I-Q architecture introduced in Sec. 4.3.1, both of which are plotted in orange in Fig. 4.4.

considered in Fig. 4.5, but now at higher SNR $\gamma_R = \gamma_D = 13$ dB, suggesting a stronger correlation between Y_R and Y_D. We note that at higher SNR, as theory suggests, the rate allowed by higher-order modulation is greater and the performance gap between different modulation schemes is larger. At high rates, for each of the modulation schemes considered, our neural CF schemes once again match the communication rate bound for perfect relay ($R \to \infty$), mirroring the trend observed in Fig. 4.5.

Considering next a complex-valued communication scenario, we illustrate the results obtained with 4-QAM and 16-QAM modulations in Fig. 4.7, where we set

$\gamma_R = \gamma_D = 7$ dB. For this case, we use an adapted version of C_{CF} from (4.4) that considers instead a complex-valued PRC. For the 16-QAM results depicted, we select the best-performing variant for the marginal model (either the joint I-Q scheme shown in Fig. 4.2a or the split I-Q version, both of which are introduced in Sec. 4.3.1). Consistent with the trends observed in Figs. 4.5 and 4.6, our neural CF scheme again meet the respective perfect relay baselines ($R \to \infty$). These empirical results further confirm that our learning-based relay compression schemes can be easily adapted to any chosen fixed modulation, scoring higher communication throughput as the order of modulation increases.

4.4.3 Interpretability of the Learned CF Relaying Schemes

The maximum a posteriori (MAP) estimator for W in the PRC of Fig. 4.1 is as follows:

$$\hat{w} = \arg\max_{w} p(w|y_D, u), \tag{4.16}$$

$$= \arg\max_{w} p(y_D|w)\ p(u|w)\ p(w), \tag{4.17}$$

where we have used the independence of y_D and u given w. For equally likely symbols, $p(w)$ is a constant and therefore, can be removed from (4.17). Note that the term $p(u|w)$ represents the likelihood of w based on the relay's quantized observation u, and it *updates* the destination's likelihood $p(y_D|w)$ in (4.17). For reference, without the relay, the optimal decision thresholds on Y_D for the maximum likelihood estimator for a PAM (QAM) modulation under Gaussian noise would be the intersection between adjacent likelihoods, that is the middle point (line) between adjacent symbols [97].

We recall that for our learned CF schemes introduced in Sec. 4.3, $u = \mathrm{e}_\theta(y_R)$, and the posterior estimated by the neural demodulator is $\mathrm{p}_\phi(w|y_D, \mathrm{e}_\theta(y_R))$. In the remainder of this section, we provide results that help to visualize and interpret the quantization boundaries recovered by the neural encoder e_θ and learned MAP (4.10) decision thresholds adopted by the demodulator. First, we show that the marginal CF variant (Fig. 4.2a) groups the quantized indices at the relay, by assigning the same quantization index to discontiguous intervals in the source space. This empirical evidence suggests that the scheme effectively uses the side information Y_D during compression. Next, we show how the relay's likelihood $p(\mathrm{e}_\theta(y_R)|w)$ operationally shifts the decision thresholds.

Fig. 4.8 illustrates the marginal CF scheme and the demodulation's hard decision regions (see (4.10)) for 4-PAM with $\gamma_D = \gamma_D = 13$ dB and relay rate of $R \approx 1$. The vertical axis and horizontal axis show Y_R and Y_D, respectively. The colors represent the transmitted indices $\mathrm{e}_\theta(Y_R)$ by the relay, and the horizontal lines are the corresponding quantization boundaries. Note that this neural CF architecture exhibits binning (grouping) since non-adjacent intervals are assigned to the same index (same color). It is worth noting that this recovered grouping behavior is similar to the random binning operation in the achievability proof of the WZ theorem [69] and also in the achievability of CF [68]. This emergence of learned one-shot binning behavior also explains the further reduction in relay rate compared to the point-to-point model, as illustrated in the experimental results shown in Figs. 4.3 and 4.4. Unlike the marginal scheme, the point-to-point model (Fig. 4.2c), however, lacks access to the side information signal Y_D, which is available at the decoder, during compression. Therefore, this latter model cannot learn a binning behavior in the relay compressor (not depicted). In contrast, the conditional variant

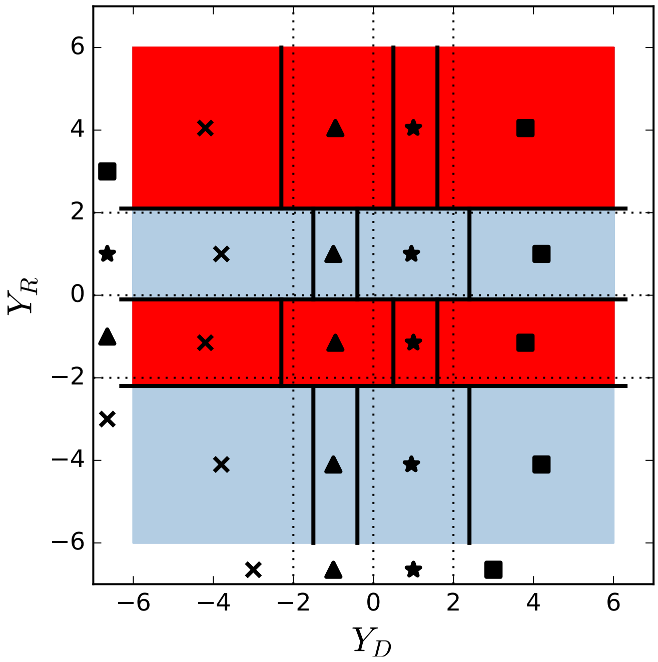
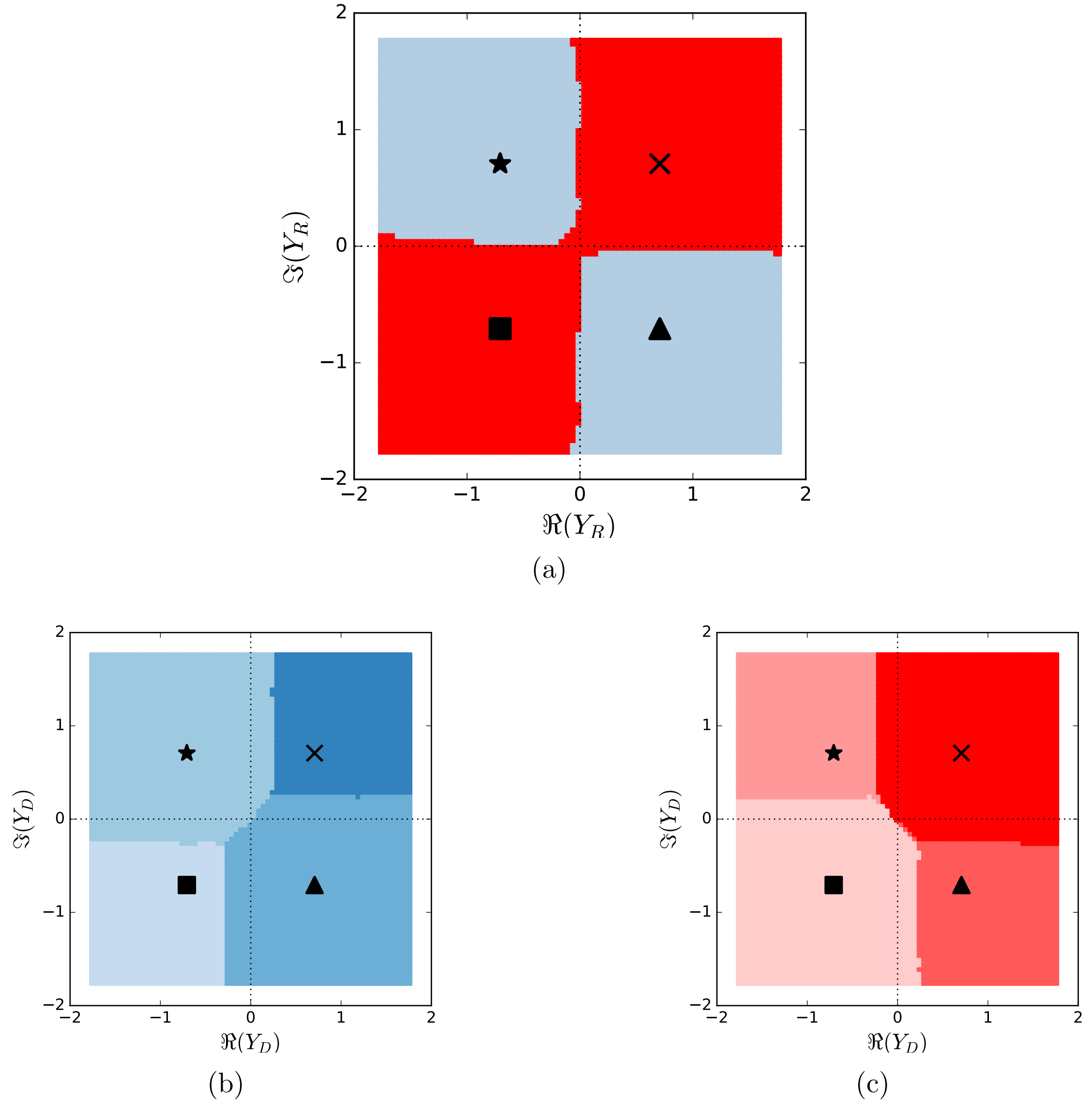


Figure 4.8: Visualization (best viewed in color) of the learned CF strategy (marginal scheme in Fig. 4.2a) and demodulation decisions for the 4-PAM modulation with $\gamma = 13$ and relay rate $R \approx 1$. The horizontal lines denote the quantization boundaries on Y_R, and the colors designate the transmitted index $\mathrm{e}_\theta(Y_R)$. The vertical lines denote the hard decision boundaries for the demodulator, and the markers represent the decisions. The transmitted symbols (denoted by cross, triangle, star, square) are also reported near the axis for reference.

(Fig. 4.2b) leverages the side information not only during compression but also within the entropy coding stage. This enables the conditional scheme to execute binning over long sequences i.e., in a multi-shot fashion. Note that such a high-order binning scheme, facilitated by the SW coder, is inherently more efficient than the one-shot binning achievable by an encoder at the relay. As the model e_θ compresses each source realization one at a time, it can only bin the quantized indices at the relay in a one-shot fashion.

The vertical lines in Fig. 4.8 denote the hard decision boundaries, where the markers denote the decisions $\hat{W}$. We observe that the decision boundaries are shifted with respect to the midpoints between transmitted symbols (optimal boundaries without relaying). This highlights the interpretability of our neural CF relaying scheme. For example, when *cross* or *star* are transmitted, the index *blue* will be the (most likely) relayed index. In this case, the decision regions for *cross* and *star* at the destination are larger than the other symbols.

Fig. 4.9 shows the learned marginal CF strategy for the complex-valued 4-QAM modulation when $\gamma_D = \gamma_R = 7$ dB and relay rate of $R \approx 1$. The vertical and horizontal axis of each subfigure represent real and imaginary parts of Y_R and Y_D. Fig. 4.9a reports the output of the relay's encoder, where the color represents $\mathrm{e}_\theta(Y_R)$. One can note that the regions surrounding the farthest symbols are paired with the same encoding (color) $\mathrm{e}_\theta(Y_R)$. Similar to Fig. 4.8, one can argue that this is yet another instance of binning in the relay compressor. Fig. 4.9b shows the hard decision boundaries on Y_D when $\mathrm{e}_\theta(Y_R)$ corresponds to the *blue* index. Meanwhile, Fig. 4.9c shows the hard decision boundaries on Y_D when $\mathrm{e}_\theta(Y_R)$ corresponds to the *red* index. Again, the decision boundaries are shifted to favor the symbols that were most likely to be received at the relay.



(a)

(b)

(c)

Figure 4.9: Visualization (best viewed in color) of the learned CF strategy (marginal scheme in Fig. 4.2a) and demodulation decisions for the 4-QAM modulation with $\gamma = 7$ dB and relay rate $R \approx 1$. Figure (a) shows the quantization boundaries on Y_R (on the complex plane), and the colors designate the transmitted index $\mathrm{e}_\theta(Y_R)$. Figures (b) and (c) show the hard decision boundaries for the demodulator as a function of Y_D (on the complex plane), where different colors represent the different decisions. Figure (b) represents the decisions when $\mathrm{e}_\theta(Y_R)$ corresponds to the *blue* index from Figure (a); Figure (c) represents the decisions when $\mathrm{e}_\theta(Y_R)$ corresponds to the *red* index from Figure (a). The transmitted symbols (denoted by cross, triangle, star, square) are also reported for reference.

In practice, Figs. 4.8 and 4.9 can be used as look-up tables for direct deployment of the resulting CF relaying strategies, including both the relay's encoder and the destination's demodulator. Although ANN-based architectures (Fig. 4.2) were used to minimize the loss function in (4.12), the actual CF scheme and the hard demodulator implementation at test time rely only on the learned quantization boundaries and threshold values shown in Figs. 4.8 and 4.9.

4.4.4 Robustness to Signal-to-Noise Ratio (SNR) Variations

In Sections 4.4.1, 4.4.2, and 4.4.3, we evaluated the performance at the same SNRs used for training. In this section, we analyze robustness with respect to the training SNR. We consider 4-PAM modulation for the source X, and a range of test SNRs $\gamma_D, \gamma_R \in \{0, 1, \dots, 6\}$ dB. We consider models that satisfy the relay rate constraint of $R \lessapprox 1$. Note that for this SNR range, it is known that the 4-PAM capacity is superior to the BPSK one, and almost equivalent to those achieved by higher order PAM modulations [97]. This is also evident in Fig. 4.5 at $\gamma_D = \gamma_R = 3$ dB. Adapting the modulation order to the SNR is a key component of modern communication systems relying on link adaption [98], and as such, we assume 4-PAM modulation is only used in the above SNR range.

We study the following test scenarios:

1. Same SNR at both the relay and the destination, i.e., $\gamma_D = \gamma_R = \gamma \in \{0, 1, \dots, 6\}$ dB;

2. Relay SNR fixed at $\gamma_R = 3$ dB, and variable destination SNR $\gamma_D \in \{0, 1, \dots, 6\}$ dB;

3. Destination SNR fixed at $\gamma_D = 3$ dB, variable relay SNR $\gamma_R \in \{0, 1, \dots, 6\}$

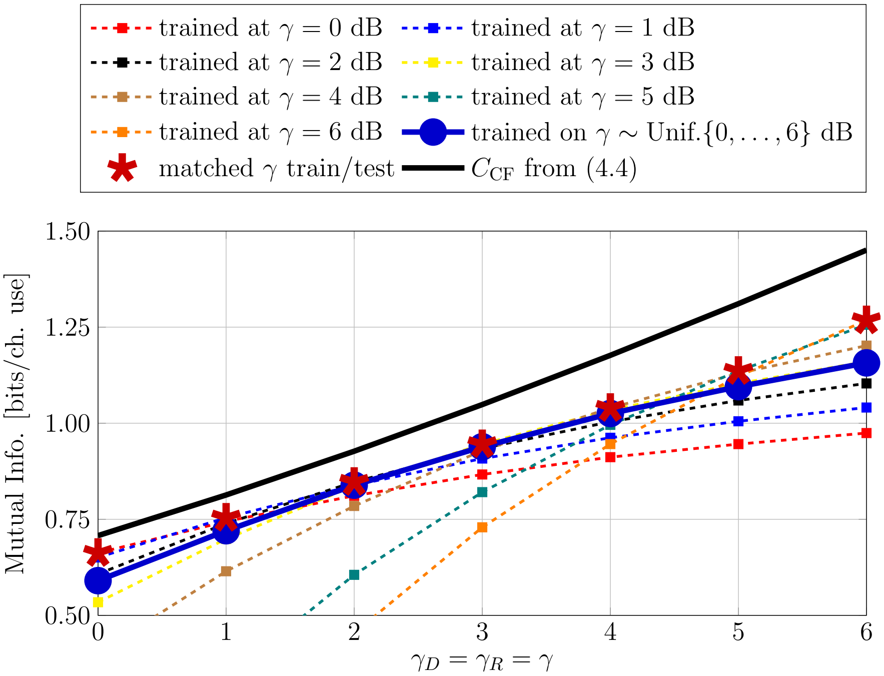


Figure 4.10: Robustness analysis when the destination and the relay have the same test signal-to-noise ratio (SNR) $\gamma_D = \gamma_R = \gamma$. The rate constraint is $R \approx 1$ for all points. The lines represent the mutual information obtained with the learned CF strategy (marginal scheme in Fig. 4.2a), as a function of the testing SNR γ. The dotted lines represent models trained for a single value of γ. The solid blue line represents the model trained for *robustness* over the SNR range of interest, i.e., the training SNR is $\gamma \sim$ Unif.$\{0, 1, \ldots, 6\}$ dB. The red stars represent the points where testing and training SNR match.

dB.

For all of the tests above, we consider baseline models that are trained at a single SNR $\gamma_D = \gamma_R = \gamma$, where $\gamma \in \{0, 1, \ldots, 6\}$ dB. In the remainder of this subsection, we analyze the performance of the baselines on the abovementioned scenarios, and propose alternative learning strategies based on robust training.

4.4.4.1 Testing at the same SNR at both the relay and the destination

Fig. 4.10 shows the mutual information when the same SNR is experienced at both the destination and the relay, i.e., $\gamma_D = \gamma_R = \gamma$. The rate constraint is satisfied for all the models $R \approx 1$ (not shown here). We also include a *robust* model trained on a range of SNRs $\gamma_D = \gamma_R = \gamma \sim$ Unif.$\{0, 1, \ldots, 6\}$ dB. Note that the baseline models trained at a single SNR perform well for adjacent SNRs too. The robust model trained on the range $\gamma \in \{0, 1, \ldots, 6\}$ dB exhibits a good compromise, offering performance similar to the model trained for the SNR in the middle of the range, and minimal performance degradation in the lower and higher end of the SNR range. Another observation is that models trained for lower SNRs exhibit less degradation at higher SNRs compared to the opposite case; in fact, the models trained at high SNRs fail at lower SNRs.

4.4.4.2 Testing on a range of SNRs at the destination, fixing the SNR at the relay

Fig. 4.11 shows the mutual information achieved as a function of the SNR at the destination $\gamma_D \in \{0, 1, \ldots, 6\}$ dB, when the SNR at the relay is fixed as $\gamma_R = 3$ dB. In other words, the statistics of the relay's received signal do not change, while the received signal Y_D at the destination has variable SNR levels. We also include two robust models, one trained for a single $\gamma_R = 3$ dB, and a range of $\gamma_D \sim$ Unif.$\{0, 1, \ldots, 6\}$ dB, and another trained for SNRs $\gamma_D = \gamma_R = \gamma \sim$ Unif.$\{0, 1, \ldots, 6\}$ dB. We note that the performance of the model trained on a range (with the same SNR on both $\gamma_D = \gamma_R = \gamma$) is equivalent to the performance of the robust model trained for $[\gamma_R = 3$ dB, $\gamma_D \in \{0, 1, \ldots, 6\}$ dB$]$.

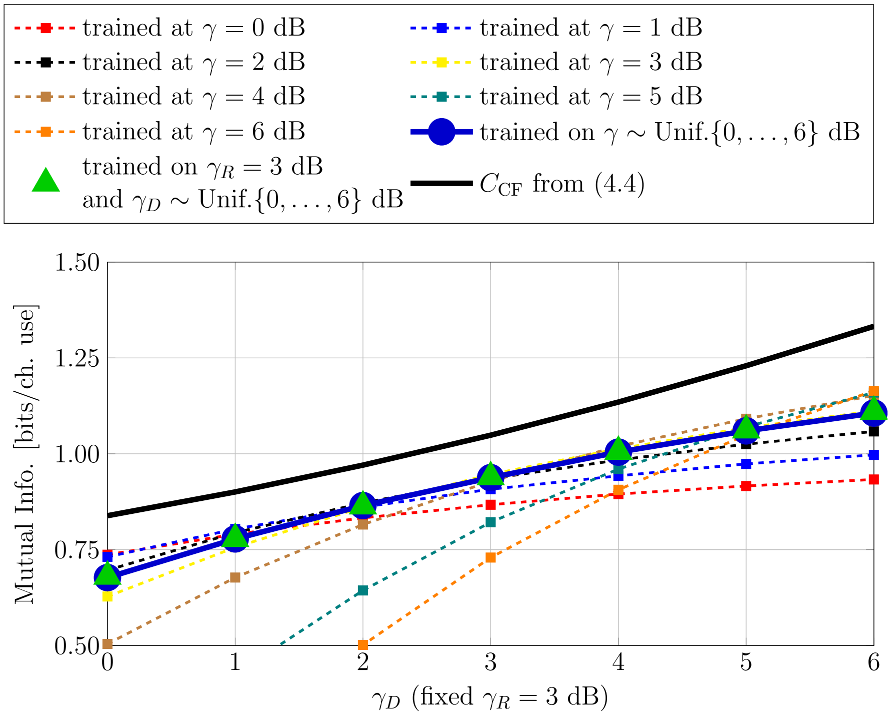


Figure 4.11: Robustness analysis when the relay signal-to-noise ratio (SNR) is fixed $\gamma_R = 3$ dB, and the destination SNR changes $\gamma_D \in \{0, 1, \ldots, 6\}$ dB. The lines represent the mutual information obtained with the learned CF strategy (marginal scheme in Fig. 4.2a), as a function of the destination SNR γ_D. The dotted lines represent models trained for a single value of $\gamma_D = \gamma_R = \gamma$. The solid blue line represents the model trained over equal SNR at both the relay and the destination $\gamma_D = \gamma_R = \gamma \sim$ Unif.$\{0, 1, \ldots, 6\}$ dB. The green triangles represent the model trained for a fixed SNR at the relay $\gamma_R = 3$ dB, and variable SNR at the destination $\gamma_D \sim$ Unif.$\{0, 1, \ldots, 6\}$ dB.

4.4.4.3 Testing on a range of SNRs at the relay, fixing the SNR at the destination

Fig. 4.12 illustrates the mutual information as a function of the SNR at the relay $\gamma_R \in \{0, 1, \ldots, 6\}$ dB, when the SNR at the destination is fixed as $\gamma_D = 3$ dB. In this case, the received signal Y_D at the destination has fixed statistics, while the relay's received signal is subjected to different SNR levels. As above, we include two robust models, one trained for a single $\gamma_D = 3$ dB, and a range of $\gamma_R \sim$ Unif.$\{0, 1, \ldots, 6\}$

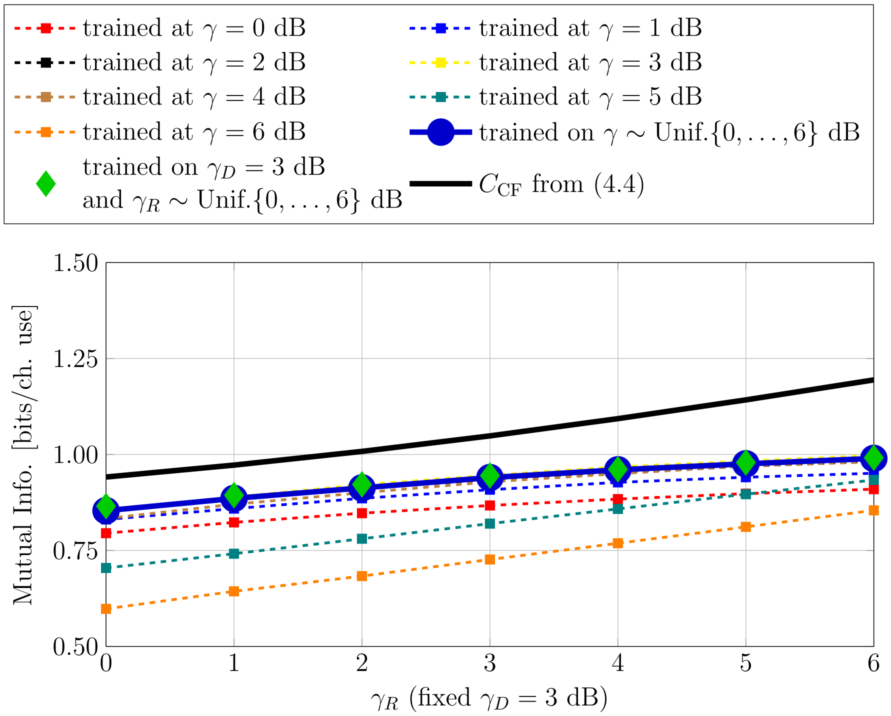


Figure 4.12: Robustness analysis when the destination signal-to-noise ratio (SNR) is fixed $\gamma_D = 3$ dB, and the relay SNR changes $\gamma_R \in \{0, 1, \ldots, 6\}$ dB. The lines represent the mutual information obtained with the learned CF strategy (marginal scheme in Fig. 4.2a), as a function of the relay SNR γ_R. The dotted lines represent models trained for a single value of $\gamma_D = \gamma_R = \gamma$. The solid blue line represents the model trained over equal SNR at both the relay and the destination $\gamma_D = \gamma_R = \gamma \sim$ Unif.$\{0, 1, \ldots, 6\}$ dB. The green diamonds represent the model trained for a fixed SNR at the destination $\gamma_D = 3$ dB, and variable SNR at the destination $\gamma_R \sim$ Unif.$\{0, 1, \ldots, 6\}$ dB.

dB, and another one trained for SNRs $\gamma_D = \gamma_R = \gamma \sim$ Unif.$\{0, 1, \ldots, 6\}$ dB. Similar to the previous scenario, the performance of the model trained on a range (with the same SNR on both $\gamma_D = \gamma_R = \gamma$) is equivalent to the performance of the robust model trained for [$\gamma_D = 3$ dB, $\gamma_R \in \{0, 1, \ldots, 6\}$ dB]. We also note that, in this scenario, the baseline models trained at an SNR in the vicinity of $\gamma_D = \gamma_R = \gamma = 3$ dB perform well.

In summary, we showed that training on a range of *equal* SNRs for both at the relay and the destination provides a good compromise in performance. This suggests good generalization capabilities for both the compressor and the demodulator, eliminating the need for ad-hoc SNR choices during training. Experimental results suggest that knowing the SNR at the destination is generally more important in order to achieve good performance. In principle, the destination could have a fine-tuned model for each SNR (or SNR range) it experiences. Concurrently, the experiments demonstrate that training robust relay nodes only requires a rough estimate of the SNR range at the relay.

4.5 Summary

In this work, we have revisited CF relaying in the context of learned distributed compression and incorporated a task-oriented neural WZ compressor into a PRC setup as a practical form of CF relaying mechanism. Our proposed framework represents the first proof-of-concept work for an interpretable learned CF relaying scheme, where both the compressor and the demodulator components are parameterized with lightweight ANNs. Such a design choice also enables us to provide post-hoc explanations of these learned components by explicitly visualizing their behaviors.

Our results demonstrate that the learned CF schemes exhibit characteristics of the optimal asymptotic CF, such as binning of the quantized indices at the relay. We also note that the performance of these schemes, across various modulation schemes (both real and complex-valued), meets the communication rate of perfect relay ($R \to \infty$) with minimal relay rate R. We have also demonstrated that training over a range of SNRs, both at the destination and the relay, provides good generalization over the range of interest, with minimal performance degradation compared to models trained for a specific SNR.

Extending our framework to a general relay channel, in which the destination does successive decoding of the compressed relay index and the source information, would be possible. Additional design constraints arising from incorporating a learned CF in full-duplex and half-duplex relay channels, as well as more complex and realistic channel models would be interesting future research directions. Another promising area for future exploration is extending the proposed neural CF frameworks to handle multi-hop networks and MIMO relay channels.

Chapter 5

Conclusion and Future Work

In this thesis, we proposed two instances of task-aware design of communication systems for general-purpose networks. In Chapter 3, we showcased our precoding-oriented CSI feedback strategy for multi-cell multi-user systems, which can help in unlocking the potential of massive MIMO systems. In Chapter 4, we provided the first proof-of-concept for interpretable CF relaying schemes. For specific conclusions about these two problems, we refer the reader to the summary sections at the end of the respective chapters, namely Section 3.5 and Section 4.5, respectively.

In both scenarios explored in this thesis, we demonstrated that learned neural compression methods, combined with a carefully formulated loss function, enabled the end-to-end optimization of the aforementioned communication problems. One of our main contributions consists of designing loss functions that directly align with the key performance metrics of the real-world communication system. In particular, the tunable nature of the proposed loss functions enables the exploration of various tradeoffs between communication overhead and end-to-end performance, without requiring any explicit changes to the underlying neural architecture. Furthermore, the flexibility of our approach is emphasized by the fact that while the specific neural architecture is chosen based on the nature of the data being processed, the optimization objective and training strategy are adaptable to a wide range of neural network choices. This flexibility allows the work presented in this thesis to be easily extended to other communication scenarios involving different channel models.

Some future research directions for the multi-cell precoding-oriented CSI feedback approach are as follows. One possibility is to analyze the performance of our approach performance across a broader range of channel models, including real-world data, to validate its effectiveness in practical scenarios. Additionally, extending the framework to a MIMO-OFDM setting could prove valuable, as OFDM is a widely used modulation scheme in modern wireless systems. Finally, conducting comprehensive system-level simulations would provide a deeper understanding of the approach's performance in complex real-world network environments.

Regarding the detection-oriented neural relays, several future research directions are envisioned here. One direction is to extend the current framework to the more general relay channel, including more realistic channel models that incorporate factors like fading and interference. Additionally, extending our approach to half- and full-duplex channels would be valuable, as it would address its applicability in various real-world scenarios. Another potential direction is to explore the application of detection-oriented neural relays in MIMO relay channels, which would be of great interest to next-generation wireless systems. Finally, investigating the scalability of these learned relays in multi-hop and denser networks could help in the network design and optimization.

Bibliography

[1] Y. Shkel, M. Raginsky, and S. Verdú, “Universal lossy compression under logarithmic loss,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, June 2017, pp. 1157–1161.

[2] V. Goyal, “Theoretical foundations of transform coding,” *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 9–21, 2001.

[3] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression,” in *International Conference on Learning Representations*, 2017.

[4] F. Sohrabi, K. M. Attiah, and W. Yu, “Deep learning for distributed channel feedback and multiuser precoding in FDD massive MIMO,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4044–4057, 2021.

[5] O. Simeone, E. Erkip, and S. Shamai, “On codebook information for interference relay channels with out-of-band relaying,” *IEEE Transactions on Information Theory*, vol. 57, no. 5, pp. 2880–2888, 2011.

[6] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[7] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press, 1949.

[8] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[9] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[10] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.

[11] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2595–2621, 2018.

[12] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.

[13] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.

[14] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Y. Li, "Semantic communications: Principles and challenges," *arXiv preprint arXiv:2201.01389*, 2021.

[15] G. Shi, Y. Xiao, Y. Li, and X. Xie, “From semantic communication to semantic-aware networking: Model, architecture, and open problems,” *IEEE Communications Magazine*, vol. 59, no. 8, pp. 44–50, 2021.

[16] C. Chaccour, W. Saad, M. Debbah, Z. Han, and H. V. Poor, “Less data, more knowledge: Building next generation semantic communication networks,” *arXiv preprint arXiv:2211.14343*, 2022.

[17] X. Luo, H.-H. Chen, and Q. Guo, “Semantic communications: Overview, open issues, and future research directions,” *IEEE Wireless Communications*, vol. 29, no. 1, pp. 210–219, 2022.

[18] W. Yang, H. Du, Z. Q. Liew, W. Y. B. Lim, Z. Xiong, D. Niyato, X. Chi, X. Shen, and C. Miao, “Semantic communications for future internet: Fundamentals, applications, and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 213–250, 2023.

[19] A. Gersho and R. M. Gray, *Vector quantization and signal compression.* Kluwer Academic Publishers, 1991.

[20] F. Carpi, S. Garg, and E. Erkip, “Single-shot compression for hypothesis testing,” in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2021, pp. 176–180.

[21] J. N. Tsitsiklis, “Decentralized detection by a large number of sensors,” *Mathematics of Control, Signals and Systems*, vol. 1, no. 2, pp. 167–182, Jun 1988. [Online]. Available: https://doi.org/10.1007/BF02551407

[22] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series*

in Telecommunications and Signal Processing). USA: Wiley-Interscience, 2006.

[23] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. USA: W. H. Freeman & Co., 1990.

[24] K. Wei, R. Iyer, S. Wang, W. Bai, and J. Bilmes, “Mixed robust/average submodular partitioning: Fast algorithms, guarantees, and applications,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS’15. Cambridge, MA, USA: MIT Press, December 2015, p. 2233–2241.

[25] A. No, “Universality of logarithmic loss in fixed-length lossy compression,” *Entropy*, vol. 21, no. 6, June 2019.

[26] C.-K. Wen, W.-T. Shih, and S. Jin, “Deep learning for massive MIMO CSI feedback,” *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.

[27] G. Wallace, “The jpeg still picture compression standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[28] J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici, “Nonlinear transform coding,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 339–353, 2021.

[29] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function,” *Neural Networks*, vol. 6, no. 6, pp. 861–867, Jan. 1993. [Online]. Available: https://doi.org/10.1016/S0893-6080(05)80131-5

[30] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations*, 2018.

[31] F. Carpi, S. Venkatesan, J. Du, H. Viswanathan, S. Garg, and E. Erkip, "Precoding-oriented massive mimo csi feedback design," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 4973–4978.

[32] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, 2010.

[33] E. Björnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: Ten myths and one critical question," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 114–123, 2016.

[34] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: A new look at interference," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 9, pp. 1380–1408, 2010.

[35] E. Björnson, M. Bengtsson, and B. Ottersten, "Optimal multiuser transmit beamforming: A difficult problem with a simple solution structure [lecture notes]," *IEEE Signal Processing Magazine*, vol. 31, no. 4, pp. 142–148, 2014.

[36] C. Windpassinger, R. Fischer, T. Vencel, and J. Huber, "Precoding in multiantenna and multiuser communications," *IEEE Transactions on Wireless Communications*, vol. 3, no. 4, pp. 1305–1316, 2004.

[37] D. J. Love, R. W. Heath, V. K. N. Lau, D. Gesbert, B. D. Rao, and M. Andrews, “An overview of limited feedback in wireless communication systems,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 8, pp. 1341–1365, 2008.

[38] X. Rao and V. K. N. Lau, “Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems,” *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3261–3271, 2014.

[39] H. Weingarten, Y. Steinberg, and S. Shamai, “The capacity region of the gaussian multiple-input multiple-output broadcast channel,” *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 3936–3964, 2006.

[40] O. Somekh, B. M. Zaidel, and S. Shamai, “Sum rate characterization of joint multiple cell-site processing,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4473–4497, 2007.

[41] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006.

[42] G. Caire and S. Shamai, “On the achievable throughput of a multiantenna gaussian broadcast channel,” *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1691–1706, 2003.

[43] M. Costa, “Writing on dirty paper (corresp.),” *IEEE Transactions on Information Theory*, vol. 29, no. 3, pp. 439–441, 1983.

[44] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, “Overview of deep learning-based CSI

feedback in massive MIMO systems," *IEEE Transactions on Communications (Early Access)*, 2022.

[45] X. Lin, "An overview of 5g advanced evolution in 3gpp release 18," *IEEE Communications Standards Magazine*, vol. 6, no. 3, pp. 77–83, 2022.

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.

[47] M. B. Mashhadi, Q. Yang, and D. Gündüz, "Distributed deep convolutional compression for massive MIMO CSI feedback," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2621–2633, 2021.

[48] M. Chen, J. Guo, C.-K. Wen, S. Jin, G. Y. Li, and A. Yang, "Deep learning-based implicit CSI feedback in massive MIMO," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 935–950, 2022.

[49] F. Carpi, S. Venkatesan, J. Du, H. Viswanathan, S. Garg, and E. Erkip, "Precoding-oriented massive MIMO CSI feedback design," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 4973–4978.

[50] J. Yang, W. Zhu, S. Sun, X. Li, X. Lin, and M. Tao, "Deep learning for joint design of pilot, channel feedback, and hybrid beamforming in fdd massive MIMO-OFDM systems," *IEEE Communications Letters*, vol. 28, no. 2, pp. 313–317, 2024.

[51] A. van den Oord, O. Vinyals, and k. kavukcuoglu, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus,

S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[52] J. Guo, C.-K. Wen, and S. Jin, “Deep learning-based csi feedback for beamforming in single- and multi-cell massive mimo systems,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 1872–1884, 2021.

[53] R. Bhagavatula and R. W. Heath, “Adaptive limited feedback for sum-rate maximizing beamforming in cooperative multicell systems,” *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 800–811, 2011.

[54] A. Wyner, “Shannon-theoretic approach to a gaussian cellular multiple-access channel,” *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 1713–1727, 1994.

[55] 3GPP, “Evolved universal terrestrial radio access (e-utra); physical layer procedures,” 3rd Generation Partnership Project (3GPP), Technical Specification TS 36.213, 2024. [Online]. Available: https://www.3gpp.org/DynaReport/36213.htm

[56] S. Lloyd, “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[57] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, “CompressAI: A pytorch library and evaluation platform for end-to-end compression research,” *arXiv preprint arXiv:2011.03029*, 2020.

[58] J. Ballé, S. J. Hwang, and E. Agustsson, “TensorFlow Compression: Learned data compression,” 2022. [Online]. Available: http://github.com/tensorflow/compression

[59] R. W. Heath Jr. and A. Lozano, *Foundations of MIMO Communication*. Cambridge University Press, 2018.

[60] E. Ozyilkan, F. Carpi, S. Garg, and E. Erkip, "Neural compress-and-forward for the relay channel," in *2024 IEEE 25th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) (to appear)*, 2024. [Online]. Available: https://arxiv.org/abs/2404.14594

[61] E. C. van der Meulen, "Three-terminal communication channels," *Advances in Applied Probability*, vol. 3, no. 1, p. 120–154, 1971.

[62] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity–Part I: System description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, 2003.

[63] J. Laneman, D. Tse, and G. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[64] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.

[65] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: A new look at interference," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 9, pp. 1380–1408, 2010.

[66] M. Najla, Z. Becvar, P. Mach, and D. Gesbert, "Integrating UAVs as transparent relays into mobile networks: A deep learning approach," in *2020*

IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, 2020, pp. 1–6.

[67] S.-Y. Lien, D.-J. Deng, C.-C. Lin, H.-L. Tsai, T. Chen, C. Guo, and S.-M. Cheng, “3GPP NR sidelink transmissions toward 5G V2X,” *IEEE Access*, vol. 8, pp. 35 368–35 382, 2020.

[68] T. Cover and A. Gamal, “Capacity theorems for the relay channel,” *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, 1979.

[69] A. Wyner and J. Ziv, “The rate–distortion function for source coding with side information at the decoder,” *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1 – 10, 1976.

[70] W. Kang and S. Ulukus, “Capacity of a class of diamond channels,” 2008.

[71] E. Özyılkan, J. Ballé, and E. Erkip, “Learned Wyner–Ziv compressors recover binning,” in *2023 IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 701–706.

[72] E. Ozyilkan and E. Erkip, “Distributed compression in the era of machine learning: A review of recent advances,” 2024.

[73] Y.-H. Kim, “Coding techniques for primitive relay channels,” in *Forty-Fifth Annual Allerton Conference*, 2007.

[74] M. Uppal, Z. Liu, V. Stankovic, and Z. Xiong, “Compress-forward coding with bpsk modulation for the half-duplex gaussian relay channel,” *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4467–4481, 2009.

[75] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Practical quantizer design for half-duplex estimate-and-forward relaying," *IEEE Transactions on Communications*, vol. 59, no. 1, pp. 74–83, 2011.

[76] Z. Liu, S. Cheng, A. Liveris, and Z. Xiong, "Slepian-wolf coded nested quantization (swc-nq) for wyner-ziv coding: performance analysis and code design," in *Data Compression Conference, 2004. Proceedings. DCC 2004*, 2004, pp. 322–331.

[77] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, 2003.

[78] C. Bian, Y. Shao, H. Wu, and D. Gunduz, "Deep joint source-channel coding over cooperative relay networks," 2022.

[79] C. Bian, Y. Shao, H. Wu, E. Ozfatura, and D. Gunduz, "Process-and-forward: Deep joint source-channel coding over cooperative relay networks," *arxiv preprint 2403.10613*, 2024.

[80] E. Arda, E. Kutay, and A. Yener, "Semantic forwarding for next generation relay networks," 2024.

[81] E. Ozyilkan, J. Ballé, and E. Erkip, "Neural distributed compressor does binning," in *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023. [Online]. Available: https://openreview.net/forum?id=3Dq4FZJSga

[82] E. Özyılkan, J. Ballé, and E. Erkip, "Neural distributed compressor discovers binning," 2023. [Online]. Available: https://arxiv.org/abs/2310.16961

[83] M. Stark, F. Ait Aoudia, and J. Hoydis, "Joint learning of geometric and probabilistic constellation shaping," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[84] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, p. 359–366, jul 1989.

[85] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, no. 6, pp. 861–867, Jan. 1993. [Online]. Available: https://doi.org/10.1016/s0893-6080(05)80131-5

[86] J. Rissanen and G. Langdon, "Universal modeling and coding," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 12–23, 1981.

[87] J. Li, Z. Tu, and R. Blum, "Slepian-Wolf coding for nonuniform sources using turbo codes," in *Proceedings of the 2004 Data Compression Conference (DCC 2004)*, 2004, pp. 312–321.

[88] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations*, 2017.

[89] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations*, 2018.

[90] J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici, "Nonlinear transform coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 339–353, 2021.

[91] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 5301–5310.

[92] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics).* Berlin, Heidelberg: Springer-Verlag, 2006.

[93] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: Composable transformations of Python+NumPy programs," 2018.

[94] E. J. Gumbel, "Statistical theory of extreme values and some practical applications : A series of lectures," 1954.

[95] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," 2016. [Online]. Available: https://arxiv.org/abs/1611.00712

[96] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[97] J. G. Proakis and M. Salehi, *Digital Communications.* McGraw-Hill, 2008.

[98] A. Goldsmith and S.-G. Chua, "Variable-rate variable-power MQAM for fading channels," *IEEE Transactions on Communications*, vol. 45, no. 10, pp. 1218–1230, 1997.